



UNIVERSIDADE DO MINHO
DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR

**CURSO DE LICENCIATURA EM
INFORMÁTICA DE GESTÃO**

PROJETO DE LICENCIATURA
ANO LETIVO 2019/2020 – 4º ANO

**MINDELBUS - APLICATIVO MÓVEL DE CONSULTA DE INFORMAÇÕES DE
AUTOCARROS**

Autor: Marvin Sérgio Correia Lima nº 2851

Orientador: Dr. Frederico Soares

**Mindelo
2020**

UNIVERSIDADE DO MINDELO
DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR

Marvin Sérgio Correia Lima

**MINDELBUS - APLICATIVO MÓVEL DE CONSULTA DE INFORMAÇÕES DE
AUTOCARROS – CIDADE DO MINDELO**

Trabalho apresentado à Universidade do Mindelo
como parte dos requisitos para obtenção do grau de
Licenciatura em Informática de Gestão.

Orientador: Dr. Frederico Soares

MINDELO
2020

Declaração De Responsabilidade

Declaro por minha honra, que o presente trabalho acadêmico foi elaborado por mim próprio, que o/a elaborei para este fim e que identifico devidamente todos os contributos de outros autores, indicando todas as fontes e conceitos usados, e que se encontram adequadamente identificados e citados, com observância das convenções do trabalho acadêmico em vigor.

Declaro, encontrar-me ciente de que a inclusão, neste texto, de qualquer falsa declaração terá consequências legais.

São Vicente, _____ de 2020

O Autor:

Marvin Sérgio Correia Lima

DEDICATÓRIA

Dedico este trabalho aos meus pais
por todo apoio e incentivo.

AGRADECIMENTOS

Aos meus pais Joaquim Delgado e Antónia Correia pelo incentivo, compreensão e paciência ao longo dessa longa jornada, e pelo apoio nos momentos mais difíceis.

Aos valorosos amigos que me apoiaram dando todo o apoio que precisei na vida pessoal e na caminhada académica me dando força e coragem para prosseguir.

Aos professores do curso, sempre mostrando a melhor maneira de aprender e conviver na área de informática e das TIC`s sendo sempre objetivos em suas lições e sugestões.

Ao orientador, Dr. Frederico Soares, por toda a atenção e apoio necessários para fazer o melhor possível e fosse possível a realização deste trabalho.

A todos que, de alguma forma, contribuíram para esta construção.

*“A verdadeira motivação vem de
realização, desenvolvimento pessoal,
satisfação no trabalho e reconhecimento.”*

(Frederick Herzberg)

RESUMO

O presente trabalho consiste na análise, desenho e implementação de um aplicativo para dispositivos móveis para consulta de informações sobre transporte público de autocarro. Informações estas que passam por mostrar rotas existentes, pontos de paragens, itinerários, e sugestão de rotas para se alcançar um determinado destino, com a possibilidade das mesmas serem visualizados via Google Maps e com uso de dados de posicionamento geográfico GPS para mostrar rotas e pontos mais próximos da posição atual do utilizador.

A proposta de solução incide sobre o sistema de transporte coletivo da cidade de Mindelo – São Vicente, para o qual se faz um apanhado da situação atual existente quanto ao sistema de informação ao utilizador.

Com ajuda da metodologia usada baseada em prototipação, procedimentos, tecnologias e ferramentas utilizadas para o desenvolvimento do projeto, se elaborou uma solução que conseguiu atender aos objetivos definidos, permitindo ter um aplicativo móvel que permite a consulta de informações de autocarros na cidade do Mindelo, apresentando funcionalidades que estarão disponíveis tanto em modo online, quanto em modo offline – com algumas restrições nesta última, contando como uma interface intuitiva e informativa, visando principalmente informar e guiar o utilizador desde serviço e facilitar o seu dia.

Palavras-chave: Dispositivos móveis, Transporte público, Informação, Google Maps, GPS.

ABSTRACT

The present work consists of the analysis, design and implementation of an application for mobile devices to consult information on public bus transport. This information includes showing existing routes, stops, itineraries, and suggested routes to reach a certain destination, with the possibility of them being viewed via Google Maps and using GPS geographic positioning data to show routes and stops close to the user's current position.

The proposed solution focuses on the public transport system in the city of Mindelo - São Vicente, for which an overview of the current situation regarding the user information system is made.

With the help of the methodology used, based on prototyping, procedures, technologies and tools used for the development , a solution was elaborated that managed to meet the defined objectives, allowing to have a mobile application that allows the consultation of bus information in the city of Mindelo , presenting features that will be available both online and offline - with some restrictions in the latter, counting as an intuitive and informative interface, aiming mainly to inform and guide the user of this service and facilitate their day.

Keywords: Mobile devices, Public transportation, Information, Google Maps, GPS.

ÍNDICE

1	INTRODUÇÃO	1
1.1	Enquadramento	1
1.2	Tema	2
1.3	Descrição do Problema	2
1.4	Justificativa	4
1.5	Objetivos	5
1.5.1	Objetivo geral	5
1.5.2	Objetivos específicos	5
1.6	Estrutura do trabalho	5
2	FUNDAMENTAÇÃO TEÓRICA	6
2.1	Transporte Público Coletivo	6
2.1.1	Transporte Coletivo Urbano de Autocarro	7
2.2	Sistemas de Informação Ao Utilizador (SIU)	9
2.2.1	O Processo de orientação espacial e sistemas de informação	9
2.2.2	Principais Funções dos SIU	10
2.3	Processo de procura de Informação em ponto de Paragem	12
2.3.1	Pontos de realçar de SIU na cidade do Mindelo	14
2.4	Conceitos ligados ao desenvolvimento para dispositivos móveis.	19
2.4.1	Dispositivos móveis	19
2.4.2	Desenvolvimento Mobile	19
2.4.3	Desenvolvimento para Android	20
2.4.4	Geo-Localização	22
3	METODOLOGIA DE DESENVOLVIMENTO	23
3.1	Método de Prototipação	24
3.2	Descrição âmbito do sistema	25
3.3	Levantamento de requisitos	25
3.3.1	Requisitos funcionais	26
3.3.2	Requisitos não funcionais	26

3.4	Arquitetura do sistema.....	27
3.5	Análise e modelação do sistema.....	28
3.5.1	Modelo Entidade – Relacionamento (ER).....	28
3.5.2	Diagrama de Casos de Uso.....	30
3.5.3	Diagramas de Sequência.....	32
3.5.4	Diagrama de Atividade.....	35
3.5.5	Diagrama de Classes.....	36
4	DESENVOLVIMENTO DO SISTEMA – PROJETO.....	37
4.1	Tecnologias e Ferramentas de Desenvolvimento.....	37
4.1.1	Android.....	38
4.1.2	Android Studio.....	39
4.1.3	Linguagem de Programação – JAVA.....	40
4.1.4	SQLite.....	41
4.1.5	Google Maps Platform Android API.....	42
4.1.6	Firebase.....	43
4.1.7	Astah.....	44
4.2	Equipamentos.....	45
4.3	Definição de Protótipo a se obter.....	45
4.4	Apresentação do Sistema.....	46
4.5	Implementação do Sistema - Protótipo.....	46
4.5.1	Prints de Tela e codificação.....	46
5	CONCLUSÃO.....	58
5.1	Resultados Obtidos.....	59
5.2	Dificuldades Encontradas.....	59
5.3	Desenvolvimentos futuros.....	60
	REFERÊNCIAS BIBLIOGRÁFICAS.....	61
	APÊNDICE.....	63
	APÊNDICE A – Diagrama de Classes principais da aplicação.....	63
	APÊNDICE B – Track – Tempo Real.....	64

LISTA DE FIGURAS

Figura 1. Comparativo ocupação da via com veículos privados e transporte coletivo.	7
Figura 2. Modelo de processo de orientação espacial.	9
Figura 3. Tipos de informação desejada.....	11
Figura 4. Decomposição sequencial da tarefa cognitiva de utilização do transporte público coletivo por utilizadores..	13
Figura 5. Ilustração Visual de canais de um sistema de informação ao utilizador.....	15
Figura 6. Infraestrutura de software do Android	21
Figura 7. Modelo Prototipagem.....	24
Figura 8. Arquitetura Sistema.....	27
Figura 9. Modelo Entidade Relacionamento baseado no modelo GTFS	28
Figura 10. Diagrama de Casos de Uso	30
Figura 11. Diagrama de Sequencia Rotas.....	32
Figura 12. Diagrama de Sequencia Pontos de Paragem	33
Figura 13. Calcular Sugestão Trajetos.....	34
Figura 14. Diagrama Atividade Calcular Trajeto	35
Figura 15. Diagrama de atividade encontrar paragens mais próximos do utilizador	36
Figura 16. Logotipo do Android.....	38
Figura 17- Logotipo Android Studio.....	39
Figura 18. Logotipo Java	40
Figura 19. Logotipo SQLite	41
Figura 20. Google Maps	42
Figura 21. Firebase	43
Figura 22. Logotipo Astah.....	44
Figura 23. Página Inicial Protótipo.....	46
Figura 24. Informações detalhadas das Rotas	47
Figura 25. Código parcial visualizar rotas no Mapa.....	48
Figura 26. Informações detalhadas Pontos de Paragens.....	49
Figura 27. Exemplo código para se obter o posição atual do utilizador / dispositivo	50
Figura 28. Código parcial de formula de procurar paragens próximos via GPS.....	51
Figura 29. Código parcial para procura de paragens próximos via Google Maps	52

Figura 30. Informações de Sugestão de Rotas.....	53
Figura 31. Código parcial para receber input e sugerir rota baseado no destino.....	54
Figura 32. Código parcial para sugestão de rotas via inserção de origem e destino	55
Figura 33. Código parcial sugestão de rotas via GoogleMaps	56
Figura 34. Diagrama de Classes principais da aplicação	63
Figura 35. Visualização de Rotas em Tempo Real.....	64
Figura 36. Código parcial implementação do Track Tempo Real.....	65

LISTA DE TABELAS

<i>Tabela 1. Requisitos Funcionais</i>	<i>26</i>
<i>Tabela 2. Requisitos não funcionais.....</i>	<i>27</i>
<i>Tabela 3. Fluxo de Eventos de Caso de Uso</i>	<i>31</i>
<i>Tabela 4- Ferramentas e Tecnologias Desenvolvimento</i>	<i>38</i>
<i>Tabela 5. Equipamentos necessários para desenvolver Aplicativo</i>	<i>45</i>

LISTA DE SIGLAS E ABREVIATURAS

GPS	<i>Global Positioning System.</i>
SIU	Sistemas de Informação Ao Utilizador.
URL	<i>Uniform Resource Locator.</i>
SDK	<i>Android Software Development Kit.</i>
API	<i>Application Programming Interface</i>
S.O	Sistema Operativo
UML	<i>Unified Modeling Language</i>
RF	Requisito funcional
RNF	Requisito não funcional
ARME	Agência de Regulação Multisectorial da Economia
INE	Instituto Nacional de Estatística – Cabo Verde
GTFS	<i>Google Transit Feed Specification</i>
JSON	<i>JavaScript Object Notation</i>

CAPÍTULO I

1 INTRODUÇÃO

O sistema de transporte é de grande importância quanto as demais redes de infraestrutura de qualquer região e o grau de desenvolvimento está ligado diretamente a qualidade e disponibilidade do seu sistema de transporte. Toda cidade requer mobilidade para o seu funcionamento - pessoas precisam do sistema de transporte para se locomover de suas residências para os locais de trabalho, lazer, acesso a educação e ademais, quanto mais sabendo que parte da população não tem rendimentos para obter o próprio veículo, que por sua vez não seria de todo recomendável dado ao congestionamento que provocaria.

Antes e agora informações sobre rotas de transporte urbana na cidade do Mindelo tem sido escassa, portanto, se pretenderá criar um aplicativo para dispositivos móveis com sistema operacional Android que permite fornecer informações dos transporte urbano, de maneira que o utilizador possa se interagir, guiar e se mover da melhor forma na cidade. Tendo por base o facto de a utilização de smartphones tem-se tornado cada vez mais comum, principalmente para acesso à Internet e busca de informações.

O presente documento apresenta e contextualiza os diferentes tópicos apresentando as considerações iniciais do trabalho, a justificativa e os objetivos e apresentados os capítulos subsequentes que compõem o texto.

1.1 Enquadramento

São Vicente sendo uma das ilhas com maior número de habitantes, possui entre seus serviços o de transporte público de autocarros, que tem sido de uma grande contribuição para a mobilização dos seus cidadãos.

Contudo um dos problemas presentes na utilização deste mesmo serviço de transporte é a falta de uma melhor e mais abrangente sistema de informação para o utilizador sobre o transporte público de autocarro, no que se refere as rotas existentes, pontos de paragens, horários, o que causa na maioria dos casos uma série de inconvenientes, tais como: perda de

tempo, necessidade de perguntar a terceiros, riscos de escolher uma rota incorreta ou alternativa e ou então levar a desistência de seu uso.

Portanto torna-se necessários melhorar a interação dos cidadãos com a rede de transporte coletivo, disponibilizando mais informações sobre o serviço, de forma pratica e intuitiva, tendo para isso implementado uma aplicação para plataforma Android, que sirva como um canal de informação ao utilizador, e permitindo melhorar o serviço de transporte.

1.2 Tema

Aplicativo Móvel de consulta de informações de autocarros – Cidade do Mindelo.

1.3 Descrição do Problema

O principal problema a ser abordado é a insuficiência de informações existentes e de canais de divulgação para o transporte público urbano de autocarro onde existe grande movimentação de pessoas diariamente, e que devido a falta de uma maior disponibilização de informações sobre este serviço, resulta em uma serie de inconvenientes a uma melhor prestação do serviço e consequente satisfação.

Na conceção de um sistema de informação, tende-se a encontrar dois tipos de situações, a primeira é o excesso de informações, que não tornam, necessariamente, o sistema útil e fácil de usar e a segunda é a escassez de informação, em decorrência de uma suposição que os utilizadores comuns já conhecem o sistema ou o entendem intimamente.

Situações decorrentes do problema

Dentro do conceito de transporte público eficiente e confiável, a questão da informação ao utilizador é bastante importante visto que algumas perguntas devem ser respondidas tais como:

- Qual linha devo utilizar para chegar ao meu destino?
- Quanto tempo o veículo levará para chegar ao local onde espero?
- Onde será o desembarque mais próximo do destino?

- Como chegar ao destino pretendido?
- Qual é o itinerário de certa linha?

As respostas a estas perguntas normalmente os moradores e utentes regulares deste serviço já possuem para seus trajetos rotineiros e por via de uso de referências. Entretanto, para outros utentes se levantam estas questões, sendo que principalmente para turistas estas perguntas são recorrentes na medida em que não conhecem a cidade e do problema da língua.

Consequências:

Utilizadores perguntar para familiares, amigos e conhecidos, terceiros, motorista, para a obtenção de informações, apanhar um autocarro por um tempo e depois descobrir que este não serve ao seu destino, ou que poderia realizar outro trajeto muito mais eficiente. Há possibilidade de quem até de certa forma por medo de não saber por qual linha escolher, preferir não buscá-los e aguardar sua única referência de transferência desperdiçando tempo e paciência.

Surgindo então uma possível questão, será que a questão da escolha do modo de transporte por turistas e mesmo por moradores para viagens não habituais também não está relacionada com a dificuldade de se locomover através de autocarro na cidade?

Podendo se sugerir que a escolha do modal táxi em algumas situações foi induzida pela dificuldade da obtenção de informações para a realização do trajeto.

Formulação do problema

Se pretende criar um aplicativo para que responda interrogações que em alguns momentos os cidadãos e ou turistas colocam ao querer fazer uso do transporte público:

- Existe uma guia de informações de rotas de transporte da cidade do Mindelo?
- Quais são as rotas de transporte existentes?
- Qual é a linha que desloca para onde quero ir?
- Existem mapas das linhas para que o utilizador possa se guiar melhor?
- Onde devo esperar a linha para apanhar o autocarro pretendido?

1.4 Justificativa

Tendo em conta a problemática definida, em que um dos problemas presentes no uso de autocarros é a insuficiência informações sobre o transporte público na cidade e que traz uma serie de inconvenientes, se considera pertinente o tema visto que o serviço de transporte público desempenhado um papel importante no que tange a contribuição para as necessidades de mobilidade dos cidadãos

A importância do sistema de transporte se comprova no facto de a sua utilização tem vindo a registar um crescimento. Que segundo dados da INE, na qual no quarto trimestre de 2019, o número de passageiros transportados nos autocarros em Cabo Verde aumentou 14,9%, em relação ao igual período do ano de 2018, o que representa em termos absolutos, para este mesmo período, um registo de 5.720.136 passageiros transportados nos autocarros em Cabo Verde, sendo 743.819 passageiros transportados a mais do que igual período do ano de 2018 (INE, 2020, p. 18).

Tendo em conta que em termos de serviços móveis segundo dados da ARME em que para o primeiro trimestre de 2020 o número total de cartões SIM ativos no mercado móvel em Cabo Verde foi de 593.528, representando uma taxa de penetração neste período de 107%, e que o número de acesso ao serviço de Internet via telemóvel (3G), foi de 395.047, representando uma taxa de penetração deste serviço de 71%. (ARME, 2020).

Portanto a escolha para dispositivos móveis, se torna perfeitamente viável, por ser um dos principais canais de informação, por sua crescente popularidade e se poder atingir um grande número de utilizadores e pelo facto de existir as ferramentas necessárias que permitam desenvolver a solução para tal plataforma a um baixo custo.

Por o acima exposto, justifica-se desenvolver um aplicativo móvel para dispositivos com S.O Android com o intuito de disponibilizar informações aos cidadãos das rotas de transporte urbano na cidade do Mindelo, considerando que esta deve existir e ser facultada, e desta forma poder contribuir para o crescimento tecnológico e contribuir para uma imagem inovadora da cidade usando tecnologias que são usadas em todo o mundo.

1.5 Objetivos

1.5.1 Objetivo geral

Desenvolver um aplicativo para dispositivos móveis que sirva como guia para obter informações como rotas, localização de paragens, mapas do transporte público de autocarros, informações de trajetos dos autocarros, pontos de paragem e afins na cidade do Mindelo.

1.5.2 Objetivos específicos

- Identificar as diferentes rotas existentes e as trajetórias que estás fazem;
- Mapear os pontos de paragens de autocarros na cidade do Mindelo;
- Associar as linhas e horários dos autocarros com os pontos mapeados;
- Identificar rotas e pontos de paragens mais próximos da localização do utilizador;
- Listar as informações tanto em modo online, quanto em modo offline;
- Sugerir rotas possíveis dados ponto de origem e destino, ou ponto destino pretendido;
- Disponibilizar o aplicativo no Playstore para que os utilizadores tenham o seu acesso;
- Verificar as ferramentas necessárias para a conclusão do projeto final.

1.6 Estrutura do trabalho

O presente trabalho está estruturado em 5 capítulos e será desenvolvido da seguinte forma:

- Capítulo I: Introdução – Introdução, problemática, justificativa e objetivos;
- Capítulo II: Fundamentação Teórica – Abordagem dos conceitos ligados a sistemas de informação de transportes públicos e mobilidade;
- Capítulo III: Metodologia de Desenvolvimento – Descrição, Análise dos requisitos e Arquitetura para desenvolvimento do Sistema;
- Capítulo IV: Desenvolvimento do Sistema – Ferramentas utilizados para realização do projeto e solução proposta;
- Capítulo V: Conclusão – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

CAPITULO II

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo numa primeira etapa apresenta-se um pouco dos conceitos ligados transporte público urbano, mobilidade urbano, conceitos ligados e relação com sistema de informação para utilizadores desse serviço.

Tende-se demonstrar como a comunicação com o público influi na melhoria dos serviços prestados à população.

Numa segunda etapa apresentar-se a conceitos ligados a referencial teórico tecnológico, ligados ao desenvolvimento de um aplicativo para dispositivos móveis.

2.1 Transporte Público Coletivo

Transporte é a denominação dada à movimentação de pessoas e de produtos. O ideal é que esta movimentação se dê de modo seguro, rápido, confortável, conveniente, e econômico. A constante necessidade de deslocamentos nas cidades modernas faz com que a qualidade do sistema de transporte urbano seja tão importante para a qualidade de vida da população quanto outros serviços essenciais.

Transporte público ou transporte coletivo designa um meio de transporte no qual os passageiros não são proprietários deles, e são servidos por terceiros. Os serviços de transporte público podem ser fornecidos tanto por empresas públicas como privadas. Providenciam o deslocamento de pessoas de um ponto a outro na área da cidade.

Segundo (Marques, 1998, p. 19), o transporte público constitui um serviço indispensável nas cidades, apresentando um papel social e econômico de extrema importância, pois:

- “Democratiza a mobilidade, permitindo a locomoção para aqueles que não possuem automóvel ou não podem dirigir [...]”;
- Constitui uma alternativa de transporte em substituição ao automóvel, aliviando os congestionamentos, a poluição [...]”;
- Ainda como substituto do automóvel, reduz a necessidade de investimentos na construção de vias e estacionamentos, etc. [...]”

Figura 1. Comparativa ocupação da via com veículos privados e transporte coletivo.



Fonte: Brazil (2008).¹

2.1.1 Transporte Coletivo Urbano de Autocarro

O transporte coletivo urbano existe em função das atividades e necessidades econômicas e sociais dos habitantes de uma comunidade.

Desta forma, tem-se em Cabo Verde o serviço de transporte por autocarro como um dos principais meios de transporte coletivo urbano, sendo esta a principal meio utilizada para este fim na cidade do Mindelo, e demonstrada a sua extrema importância nos dados disponibilizados na qual por exemplo no quarto trimestre de 2019, o número de passageiros transportados nos autocarros em Cabo Verde aumentou 14,9%, em relação ao igual período do ano de 2018. Em termos absolutos, nesse mesmo período, registaram-se 5.720.136 passageiros transportados nos autocarros em Cabo Verde, 743.819 passageiros transportados a mais do que em igual período do ano de 2018. (INE, 2020, p. 18).

Na cidade do Mindelo há dois operadores que operam no transporte coletivo de autocarros, sob a forma de prestação de serviço público sendo elas: a Transcor, e a Amizade.

O autocarro está integrado na configuração da maioria das cidades e como meio de transporte coletivo essencial.

¹ Disponível em <https://www.cidades.gov.br/images/stories/ArquivosSEMOB/Biblioteca/ManualBRT.pdf> Obtido em 05 de 03 de 2020.

Alguns elementos que são característicos do sistema de transporte público de autocarros, da qual se depende, em maior ou menor grau para a sua operação.

2.1.1.1 Rotas

As linhas de transporte público devem passar pelos principais locais de atração de viagem da área que é destinada a atender, assim como proporcionar uma cobertura satisfatória das áreas habitadas. Locais como zonas industriais, terminais de transportes, escolas, universidades, centros desportivos, etc., localizados na área de atendimento, são pontos de passagem quase sempre obrigatórios. O itinerário da linha deve, também, permitir que todos os habitantes, de uma determinada área coberta pelo sistema de transporte público, possam utilizá-lo com percursos realizados a pé dentro de limites toleráveis.

Em função do atendimento oferecido e do itinerário desenvolvido, as linhas de autocarro recebem uma classificação. Cada tipo de linha se ajusta melhor a um determinado padrão de atendimento. Conhecer o comportamento da procura a ser atendida é imprescindível para determinar o tipo de linha de autocarro que apresentará os melhores resultados operacionais (Marques, 1998).

2.1.1.2 Pontos de Paragem

Os locais de embarque e desembarque de passageiros dos autocarros, localizados nos passeios públicos, são chamados, de um modo geral, de pontos de paragem. Os pontos de paragem são de grande importância para a operação e imagem de um serviço de transporte público: neles os utilizadores têm o primeiro contato com a rede de transporte público, sendo fundamental que sejam localizados adequadamente.

Os pontos podem ser simples, onde todos os autocarros param, seletivos, onde em cada ponto só param algumas linhas pré-definidas.

A não sinalização dos pontos de paragem, fato algumas vezes comum em alguns locais, não prejudica diretamente os utilizadores regulares, que sabem onde os autocarros param, contudo para os utilizadores não regulares a sinalização clara dos pontos é fundamental na utilização do transporte coletivo. (Marques, 1998).

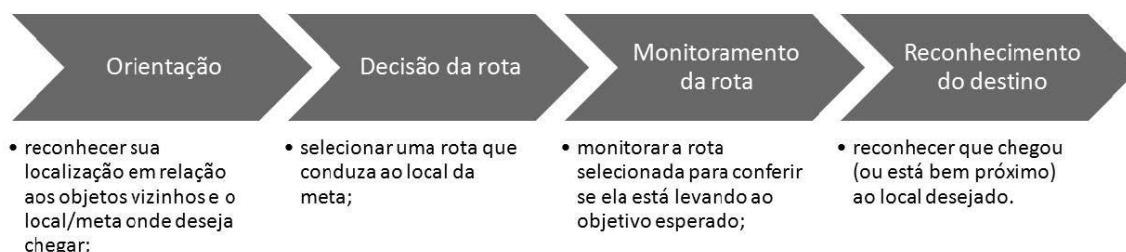
2.2 Sistemas de Informação Ao Utilizador (SIU)

Segundo (Silva, 2020, p. 34), Os sistemas de informações são uma ferramenta de diálogo com o público. Permitem extrair do conjunto de informações relativas a uma rede, aquelas que correspondem a uma necessidade específica (por exemplo: tempo de espera) ou personalizada (por exemplo: itinerário de uma linha) e ainda podem garantir um aumento da qualidade do serviço oferecido aos passageiros.

2.2.1 O Processo de orientação espacial e sistemas de informação

Segundo Satalich (1995, como citado em Lanzoni, Scariot, & Spinillo, 2011, pp. 3-4), o processo de orientação espacial de um utilizador pode ser descrito em quatro passos, conforme detalha a figura 2 a seguir:

Figura 2. Modelo de processo de orientação espacial.



Fonte: Scariot; Lanzoni; Spinillo (2011) baseado em Satalich (1995).

Compreender as etapas do processo de orientação espacial, citadas acima, permite o *design* de um sistema de informação para utilizadores de transporte público coletivo, uma vez que a presença de um sistema de informação pode auxiliar diretamente o utilizador na execução dessas etapas.

2.2.2 Principais Funções dos SIU

O objetivo de um sistema de informação para utilizadores do transporte público, para (Lanzoni, Scariot, & Spinillo, 2011), é a promoção da mobilidade, pois a informação permite que as pessoas planeiam e definam seus deslocamentos pela cidade.

Na conceção de um sistema de informação, tende-se a encontrar dois tipos de situações, a primeira é o excesso de informações, que não tornam, necessariamente, o sistema útil e fácil de usar e a segunda é a escassez de informação, em decorrência de uma suposição que os utilizadores comuns já conhecem o sistema ou o entendem intimamente.

Ainda segundo (Silva, 2020) define um sistema de informação ao utilizador pode ser usado no transporte público para exercer diferentes funções, sendo elas:

- **Promocional:** relacionada à promoção de motivos e destinos de viagens e à informação de disponibilidade dos serviços, visa melhorar a imagem do transporte público e promover sua utilização entre a população;
- **De instrução:** volta-se à função de divulgar as regras de utilização transporte público e orientar seu melhor uso;
- **Operacional:** permite que o utilizador planeia sua viagem, tenha acesso ao sistema de transporte, realize sua viagem e saiba de possíveis restrições ou alterações na programação;
- **De moderação:** atua no comportamento do passageiro, diminuindo sua ansiedade e dando a ele maior controle de escolha entre as opções disponíveis.

Segundo Molinero e Sanchez (1998, como citado em Lanzoni, Scariot, & Spinillo, 2012, pp. 4-5), um sistema de informação ao utilizador precisa atender diferentes grupos de utilizadores, que apresentam necessidades de informação variadas, de acordo com sua relação com o sistema de transporte público coletivo; sendo eles:

- **Utilizadores regulares em rota cotidiana:** devida sua maior familiaridade com o serviço, demandam menos informação sobre o uso do serviço, mas continuam

suscetíveis ao recebimento de comunicados acerca de alterações no funcionamento do sistema e das linhas de seu interesse;

- **Utilizadores regulares em rota nova:** demandam informação sobre a rede de transporte e seus itinerários;
- **Utilizadores potenciais:** necessitam de informações acerca das tarifas, horários e rotas do sistema de transporte coletivo. Ainda, as informações promocionais podem ter grande influência sobre esses utilizadores, motivando-os a utilizar o serviço; e
- **Utilizadores turistas:** desconhecem as condições locais, portanto, necessitam de informações mais completas e detalhadas sobre o serviço prestado e sobre a cidade.

Com relação ao tipo de informação, (Fernandes, 2007), analisa como sendo oito os tipos de informação desejada, detalhadas na figura 3: informações gerais sobre a rede; identificação do serviço; serviços especiais e ocasionais; itinerários; horários; tarifas; informações diversas e regras de operação.

Figura 3. Tipos de informação desejada

Informações gerais sobre a rede	<ul style="list-style-type: none"> • características gerais de médio e longo prazo, a respeito dos serviços ofertados, venda de passagens, tarifas, condições para uso, serviços da rede normal, serviços aos domingos, frequências, tempos de percurso.
Identificação do serviço	<ul style="list-style-type: none"> • informações que devem ter sua divulgação prevista pela rede, tais como: nome da parada; nome e número da linha; identificação do operador; direção/destino da linha; itinerário; e paradas.
Serviços especiais e ocasionais	<ul style="list-style-type: none"> • são aqueles não permanentes, devendo ser informados aos usuários, como linhas extras e falha no veículo.
Itinerários	<ul style="list-style-type: none"> • informações a respeito dos trajetos das linhas, podendo ser representados por ruas, mapas, esquemas.
Horários	<ul style="list-style-type: none"> • horário de chegada do ônibus na parada; horário de funcionamento da linha; horários de toda a rede; horários referentes a serviços ocasionais.
Tarifas	<ul style="list-style-type: none"> • preço das passagens, zonas tarifárias, descontos, multas e isenções.
Informações diversas	<ul style="list-style-type: none"> • Complementam outras informações: localização de pontos turísticos; números de emergência; contato para sugestões e reclamações.
Regras de operação	<ul style="list-style-type: none"> • Esclarecem direitos e deveres do passageiro; prover instruções de acessos prioritários ou gratuitos; transporte de bagagens e animais.

Fonte: Scariot; Lanzoni; Spinillo (2011) baseado em Fernandes (2007).

2.3 Processo de procura de Informação em ponto de Paragem

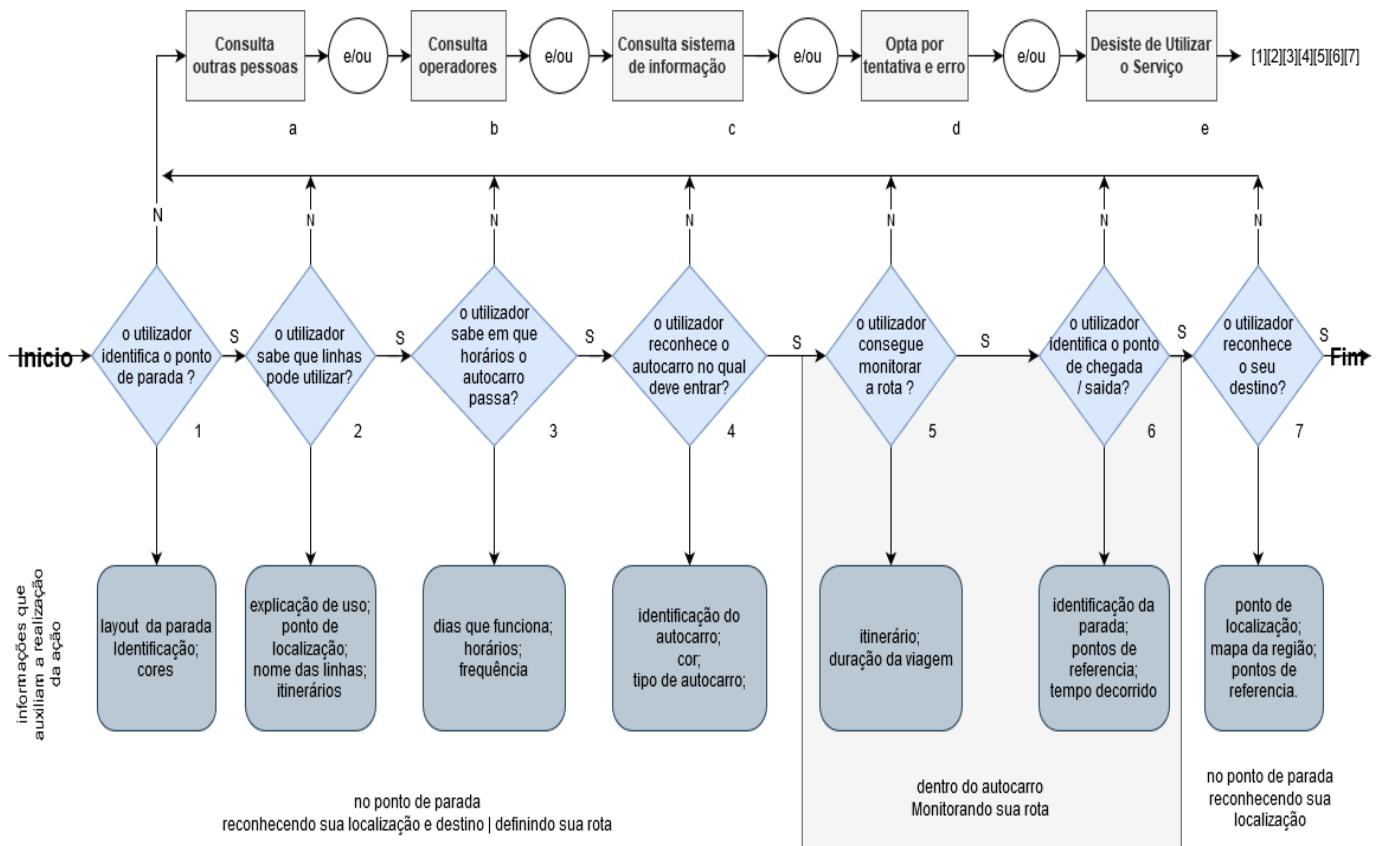
Segundo (Lanzoni, Scariot, & Spinillo, 2011), pode-se fazer a decomposição sequencial da tarefa cognitiva de utilização do transporte público coletivo, permitindo conhecer a estrutura da tarefa, suas atividades e a sequência em que essas ocorrem.

Na decomposição da tarefa do tipo sequencial, esta é desmembrada em atividades bastante específicas, representadas em um fluxograma que segue exatamente a ordem em que ocorrem. As atividades da tarefa que exigem uma ação são representadas graficamente por retângulos e aquelas relacionadas a uma tomada de decisão por um losango. Estas últimas encadeiam novas ações excludentes entre si, dependendo da resposta do utilizador, sendo: (S) Sim e (N) Não.

Ainda (Lanzoni, Scariot, & Spinillo, 2011), A sequência das atividades da tarefa acaba por seguir as etapas do processo de orientação, descrito anteriormente, de tal modo que as atividades podem ser locadas em três momentos distintos, também apontados na figura 4:

- No ponto de paragem (de embarque): onde ocorrem as quatro primeiras tomadas de decisão pelo utilizador, relacionadas ao reconhecimento de sua localização e destino, e da definição de sua rota;
- Dentro do veículo (autocarro): onde ocorrem duas tomadas de decisão pelo utilizador relacionadas ao monitoramento de sua rota;
- No ponto de paragem (de desembarque): onde ocorre a última tomada de decisão pelo utilizador, relacionada ao reconhecimento de sua localização.

Figura 4. *Decomposição sequencial da tarefa cognitiva de utilização do transporte público coletivo por utilizadores.*



Fonte: Scariot; Lanzoni; Spinillo (2011)

Desde modo (Lanzoni, Scariot, & Spinillo, 2011), Os aspetos identificados a partir da decomposição da tarefa, permitiram apontar algumas procuras de informação operacional de utilizadores do transporte público coletivo, dos utilizadores regular ao turista. Estas referem-se tanto às paragens de autocarro como aos veículos para a utilização do serviço de transporte público, sendo:

1. Identificar o mobiliário urbano como sendo um ponto de paragem;
2. Reconhecer as linhas que param naquele ponto de paragem;
3. Visualizar o itinerário completo de todas as linhas disponíveis naquela paragem;
4. Visualizar todos os pontos de paragem existentes no itinerário das linhas;
5. Reconhecer nos itinerários disponíveis o destino final de sua viagem e seu ponto de paragem de desembarque;

6. Identificar outras linhas disponíveis nas proximidades, caso não encontre uma linha favorável a si naquela paragem;
7. Identificar o tempo de espera para a chegada do veículo;
8. Relembrar e reavaliar o itinerário, após o embarque no veículo;
9. Monitorar o percurso realizado pelo veículo, para identificar se seu ponto de desembarque está próximo;
10. Planear, durante a viagem de ida, sua viagem de volta com a mesma linha.
11. Reconhecer sua localização geográfica na cidade, ao desembarcar do veículo, tendo como ponto de partida/referência a parada de autocarro, para deslocar-se até seu destino final;
12. Identificar outras fontes de informação disponíveis, caso não encontre as informações necessárias a si naquela parada ou veículo.

Estes aspetos permitam vir a ser utilizados para embasar e direccionar o planeamento e desenvolvimento de sistemas de informação para utilizadores do transporte público coletivo. Assim, podem contribuir para satisfazer as possíveis procura de informação dos utilizadores durante o processo de orientação e deslocamento pela cidade utilizando o transporte público coletivo.

2.3.1 Pontos de realçar de SIU na cidade do Mindelo

Na cidade do Mindelo existem dois operadores que operam e exploram o transporte coletivo de autocarros, sob a forma de prestação de serviço público sendo elas: a Transcor, e a Amizade.

Sendo que a Transcor S.A. é o maior operador, possuindo a maior frota e predominância na prestação deste serviço e a maior cota de mercado.

Baseado no que pode ser num dos vários canais para um sistema de informação ao utilizador, ilustrado na figura 5, se faz uma analogia com relação ao existente no maior operador, com base em dados observados no momento da elaboração deste trabalho.

Figura 5. Ilustração Visual de canais de um sistema de informação ao utilizador



Fonte: Slideshare -Sistemas de Informação ao Usuário do Transporte².

2.3.1.1 Canais Existentes – Sistema Atual

Nos veículos

- Vista eletrônica em parte dos autocarros com numeração da linha em tamanho aumentado na parte externa frontal e traseiro;
- Dentro do veículo: Anúncios sonoros informando o ponto de paragem atual.

Pontos de Paragem

- Apenas placa: sem informações quaisquer nos pontos, apenas indicativa, quanto existente.

Website

- Sem um website específico para obtenção de informação.

Smartphone

- Sem aplicativos oficiais para smartphone.

Google Maps

- Sem informações das rotas no Google Transit.

² Disponível em <https://pt.slideshare.net/RenatoArbex/sistemas-de-informao-ao-usurio-do-transporte-pblico-por-nibus-estudos-de-caso-no-brasil-e-no-mundo> Obtido em 05 de 03 de 2020.

Resumo:

- Painéis digitais nos veículos informando nº e nome da linha e resumo de zonas de passagem da rota;
- Falta de sinalização sobre trajetos, não disponibiliza busca por trajetos, linhas de autocarros e locais, sem quadro de horários, frequência, itinerário no mapa e descrito;
- Sem sistema de Informações para smartphones ou de um website específico.

2.3.1.2 Proposta de melhoria para o sistema atual através de Múltiplos Canais**Website – Internet**

- Importância de um bom website para a imagem institucional;
- Crescente expansão da Internet; Maior acesso.

Recomendações:

- Interface moderna e intuitiva;
- Possibilidade de busca por trajetos, com dados de entrada diversa, por número de linha, retorna as linhas necessárias, busca por ponto de parada (nome ou código);
- Mapa geral do trajeto, endereço dos pontos de paragens, pesquisa interativa, através de mapa, entre itinerários e paragens;
- Página detalhada para cada linha com horários, itinerário, com busca por linhas nas proximidades de um local;
- Integração das informações com o Google Transit/ Maps.

Ponto de Paragem

- Muito importante para a existência da informação na rua;
- Porta de entrada ao sistema de transporte.

Recomendações:

- Detalhamento das rotas que servem o ponto;
- Listagem das linhas;
- Mapa das proximidades do ponto;

- Um diagrama indicando o destino geral das linhas partindo daquele ponto, bem como indicando o destino das linhas da área como um todo.

Autocarros: parte Interna

- Inexistência ou baixa qualidade desta informação na cidade.

Recomendações:

- Diagrama do trajeto do autocarro com todas as paragens com seus respectivos nomes, informando as zonas em que ele passa, assim como integrações com pontos de referência dos pontos de paragem;
- Mapa geral da cidade com a sobreposição do itinerário, informando vias importantes e bairros em que a rota passa.

Smartphone

- Tendência de expansão de crescimento dos smartphones;
- Visualização da informação oficial em qualquer local.

Recomendações:

A possibilidade de busca deve ser semelhante às recomendações para o website sendo:

- Busca por linha bem como por paragem através do nome ou do código;
- Pesquisa por trajeto, informando endereço, cruzamento, ponto de referência ou zonas.

Recomendações finais:

- Investir em um excelente sistema de informação para maximizar a utilização do transporte público e melhorar sua eficiência e imagem;
- Utilização de novas tecnologias como aplicativos para smartphone e disponibilização da informação na internet.

2.3.1.3 Google Transit

O Google Transit é um serviço desenvolvido pelo Google, que utiliza o Google Maps para

mostrar informações relativas ao trânsito. Ele permite que cidades e operadores façam parceria com a empresa e que possam alimentar o sistema com informações, provendo essas informações aos utilizadores de qualquer plataforma que tenha acesso ao serviço Google Maps. Permitindo através desta selecionar origem e destino, mostrar rotas possíveis, mostra os pontos de paragem, permitindo a visualização de trechos com mais ou menos trânsito.

2.3.1.4 Dados Abertos e Limitações

Diversas cidades do mundo vêm disponibilizando seus dados na Internet, as vezes para cumprir requisitos de transparência, outras para incentivar a participação popular e inovações, ou também para a criação de novos serviços ou novos conhecimentos criados a partir da combinação de dados de diversas fontes.

Dados abertos são dados disponibilizados livremente aos quais qualquer indivíduo possa ter acesso, possa modificá-los ou compartilhá-los para qualquer finalidade, estando sujeitos no máximo a exigências que visem preservar a sua proveniência e sua abertura ³

Três leis foram propostas para os Dados Abertos Governamentais, mas é possível dizer que se aplicam a dados abertos em geral, são elas:

1. Se o Dado não pode ser encontrado e indexado na Web, ele não existe;
2. Se não estiver aberto e disponível em formato compreensível por máquina, ele não pode ser reaproveitado;
3. Se algum dispositivo legal não permitir sua replicação, ele não é útil.

Utilizando-se desses dados abertos, é possível desenvolver sistemas para processar e exibir os dados de maneira mais amigável para os utilizadores.

Para este projeto, utilizou-se dados encontrados e recolhidos do maior operador da cidade, a Transcor, dados estes datados de 2019 e das quais não se apresentavam completos.

De salientar que aquando da última revisão deste trabalho, esses dados não se encontravam mais disponíveis bem como o website não se encontrava mais disponível.

³ Disponível em <http://www.dados.gov.br/página/dados-abertos> Obtido em 01 de 05 de 2020

2.4 Conceitos ligados ao desenvolvimento para dispositivos móveis.

2.4.1 Dispositivos móveis

Atualmente a utilização de dispositivos móveis em Cabo Verde já se tornou rotineiro na vida das pessoas. De acordo com a ARME – Agencia Regulação Multisectorial, o número total de cartões SIM ativos no mercado móvel em Cabo Verde, em março de 2020 foi de 593.528, o que representa uma taxa de penetração neste período de 107%. Ainda neste mesmo período o serviço de acesso ao serviço de internet via telemóvel (3G) foi de 395.047 o que representa uma taxa de penetração deste serviço de 71%. (ARME, 2020).

2.4.1.1 Plataformas

Existem diversas plataformas de sistemas operacionais em uso nos smartphones atuais, sendo essas o meio que permite que o utilizador interage com os aplicativos e itens do hardware, como, camera, GPS, microfone, alto-falante.

As principais plataformas existentes atualmente são:

Android: sistema operacional de código aberto, e mantida e desenvolvido atualmente pelo Google, com larga utilização no mercado. O que justifica sua ampla utilização é que as marcas e os modelos que possuem esse sistema são os mais variados, de aparelhos com muitos recursos aos modelos mais básicos ou simples.

IOS: plataforma introduzida no mercado em 2007 pela Apple e de utilização exclusiva em produtos da própria marca. Por ser de uso exclusivo da marca Apple, o iOS é desenvolvido para um modelo específico de hardware, permitindo assim, que suas potencialidades sejam exploradas ao máximo. Desde sua primeira versão, foi criado para utilizar em aparelhos com telas sensíveis ao toque.

2.4.2 Desenvolvimento Mobile

O desenvolvimento de aplicações móveis se tornou uma nova oportunidade de negócio para os desenvolvedores e para as empresas, tendo em vista o crescimento exponencial do número de smartphones no mercado. Cresce na mesma proporção a busca por aplicações que satisfazem as necessidades dos utilizadores.

Essas necessidades vão de entretenimento a aplicações para uso no trabalho e que são vinculadas aos sistemas corporativos, muitas vezes de grande porte.

Segundo (Ambros, 2013), existem três tipos diferentes de aplicações mobile: híbridas, nativas e web:

- **Mobile Web App:** são aplicações via acesso com o uso de navegadores e não requerem a instalação do aplicativo no smartphone. Seu acesso é realizado por meio de um website, pela *Uniform Resource Locator* (URL). Porém, recursos como notificações do sistema operacional, acelerômetro e outros recursos de hardware não são acessíveis por essas aplicações. São, na maioria das vezes, desenvolvidos para possuírem um comportamento semelhante às aplicações nativas e utilizadas quando há necessidade de disponibilizar conteúdo ou funcionalidade em smartphones, mas que não é viável construir uma aplicação nativa.
- **Aplicativos Nativos:** são aplicativos instalados no dispositivo e adquiridos por meio de lojas, como Google Play, App Store. Seu desenvolvimento é específico para determinado operacional, podendo aproveitar todos os recursos de hardware e de software presentes no dispositivo. Além do acesso fácil aos recursos do dispositivo, se o aplicativo for bem concebido, ele respeitará os padrões de design de cada plataforma, atendo a padrões de usabilidade.
- **Aplicações Híbridas:** são aplicativos parcialmente nativos e parcialmente Web Apps. Como aplicativos nativos, eles podem aproveitar todos os recursos do smartphone e como Web Apps. Podem ser exibidos em um navegador presente no aplicativo. Essas aplicações são populares porque permitem o desenvolvimento multiplataforma, reduzindo os custos de produção.

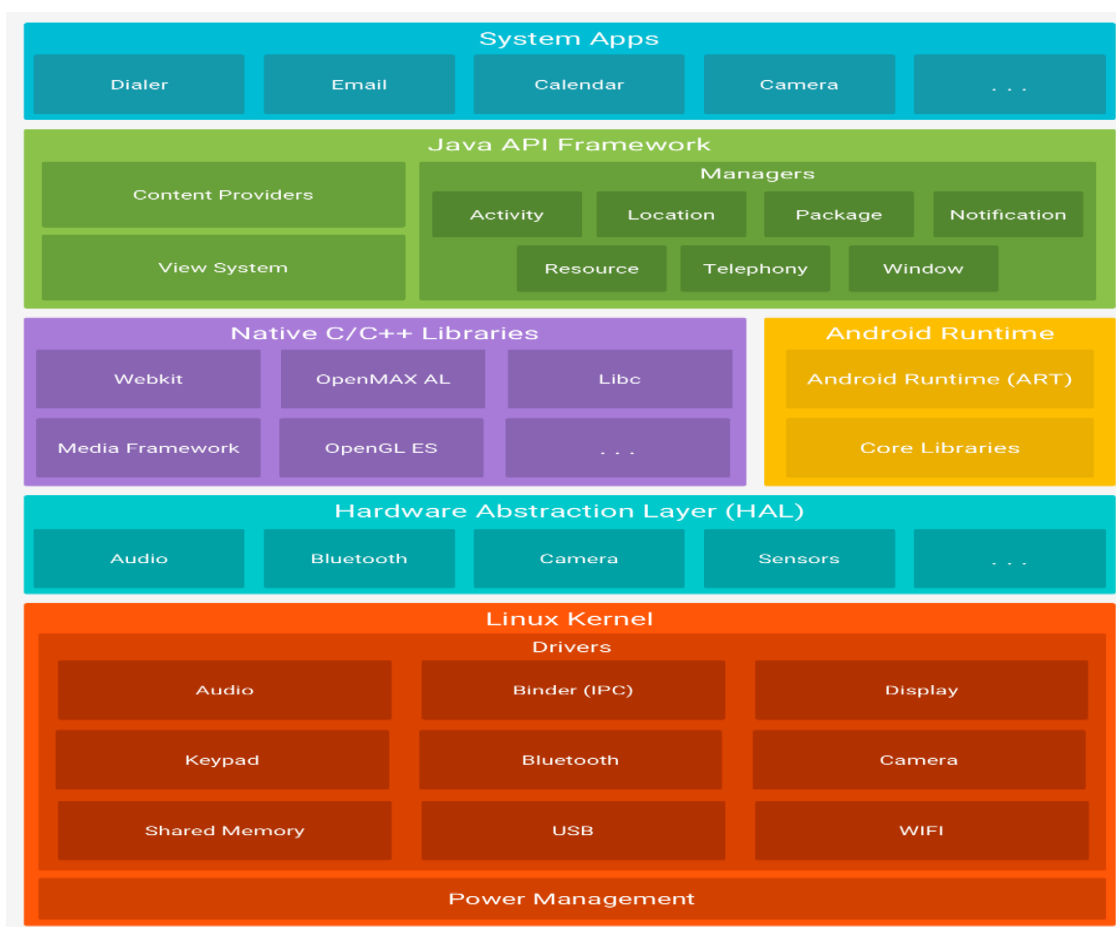
2.4.3 Desenvolvimento para Android

O Android foi criado em 2003 por Andy Rubin, Rico Miner e Nick Sears, que fundaram a empresa Android Inc. Android é um sistema operacional que tem base no kernel do Linux e seu objetivo inicial era a criação de dispositivos móveis mais inteligentes, um sistema para cameras fotográficas, mas depois foi redirecionado para ser um sistema

operacional mobile. Em 2005 a Android Inc. Foi comprada pelo Google, por 50 milhões (Hammerschmidt, 2015).

Para o desenvolvimento de aplicações, a Google disponibiliza um kit de ferramentas chamado *Android Software Development Kit* (SDK). Segundo um estudo realizado por (Ogliari & Brito, 2014), a infraestrutura do Android é composta por uma pilha que inclui um sistema operacional Linux, um conjunto de bibliotecas, uma *Application Programming Interface* (API) chamada *Android Runtime*, aplicações preexistentes no Android e aplicações diversas. Essa estrutura é apresentada na Figura 6.

Figura 6. Infraestrutura de software do Android



Fonte: Android Developer⁴

⁴ Disponível <https://developer.android.com/guide/platform?hl=pt-br> Obtido em 05 de 03 de 2020

Referindo as principais partes que compõem a infraestrutura do Android, como apresentado na Figura 5:

Kernel do Linux, Camada de abstração de hardware (HAL), *Android Runtime*, Bibliotecas C/C++ nativas, Estrutura da Java API, Aplicativos do sistema.

2.4.4 Geo-Localização

Geo-Localização é o processo de encontrar a localização exata de um computador, dispositivo de rede ou equipamento com base em medições e coordenadas geográficas (Neka, 2015). Atualmente, cada vez mais smartphones estão incluindo software e hardware de Geo-Localização.

Com a utilização dos dados de localização coletados pelo dispositivo é possível determinar, lugares, estabelecimentos, transporte público nas proximidades do utilizador por meio das informações do GPS. Os dados de localização são frequentemente utilizados por aplicações de mapas, como por exemplo:

Google Maps: permite monitorar o trânsito em tempo real, recalculando seu trajeto dinamicamente com base nos padrões de trânsito encontrados, ajudando a evitar congestionamentos (Google, 2020). Esse aplicativo oferece a possibilidade de visualizar comentários e fotos de outros utilizadores que moram ou que já estiveram no local de destino pretendido.

No Smartphone obtêm-se a posição principalmente por via da triangulação de sinal, isto é, pela rede telefônica, GPS ou GPRS ou, pela rede de dados 3G ou Wi-Fi.

CAPITULO III

3 METODOLOGIA DE DESENVOLVIMENTO

As metodologias de desenvolvimento de software permitem o planeamento correto, acompanhamento e avaliação das atividades e etapas que fazem parte do processo de desenvolvimento. Tendo em conta a filosofia que as metodologias possuem, elas podem se agrupar em dois grandes grupos: tradicionais e ágeis.

Para cumprir o objetivo principal deste trabalho, se utilizou o Modelo / Método de Prototipação. Sendo uma prototipação evolucionaria – onde se vai evoluindo a solução até a versão final do software, e a escolhe se deve a razões como os detalhados abaixo:

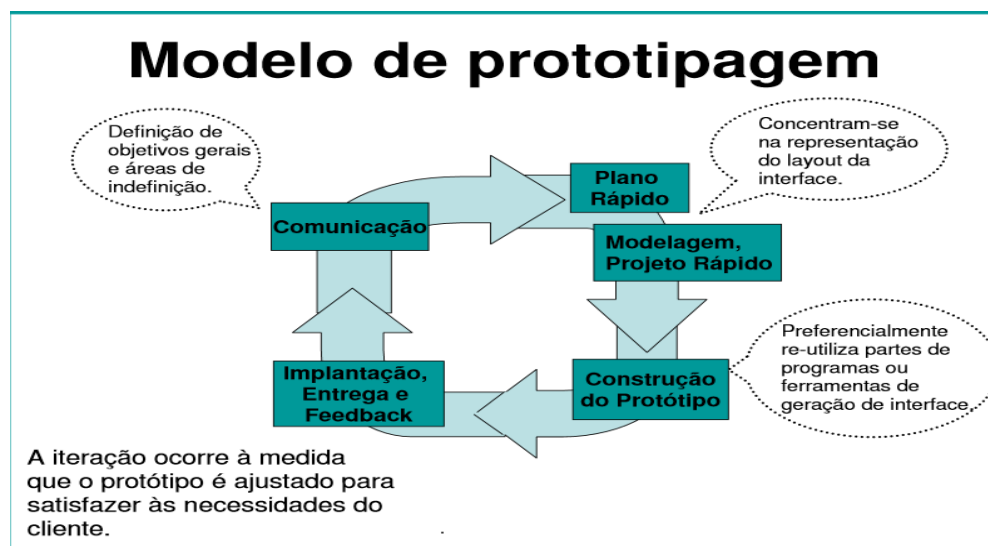
- O aplicativo se desenvolve por fases;
- Somente são desenvolvidos os requisitos bem conhecidos;
- Se faz uma implementação parcial do aplicativo e se realizam testes;
- Os utilizadores podem testar o aplicativo e adicionar novos requisitos, permitindo fazer o refinamento do produto baseado em feedback dos utilizadores;
- Permite a deteção precoce de problemas, melhorando a qualidade do produto.

Alguns problemas que podem surgir do uso deste modelo são:

- Quando o cliente testa uma versão do protótipo, este por via do seu feedback, acabar por exigir alguns acertos para fazer o seu uso;
- Codificação utilizada para apresentar o protótipo é feita utilizando o que está disponível com o objetivo de produzir rapidamente um protótipo e sendo que esta pode ser usada na versão final, mesmo não sendo a mais apropriada.

3.1 Método de Prototipação

Figura 7. Modelo Prototipagem



Fonte: (Pressman, adaptado) ⁵

Segundo (Pressman, 2011), o paradigma da prototipação começa com a comunicação que ocorre através de uma reunião com todos os envolvidos a fim de definir objetivos gerais do software e identificar quais requisitos já estão bem conhecidos e esquematizar as áreas que realmente necessitam de uma definição mais ampla.

Uma iteração de prototipação deve ser planeada rapidamente e dessa forma ocorre a modelagem na forma de um projeto rápido.

O projeto rápido foca na representação dos aspectos do software que serão visíveis aos utilizadores finais como layout da interface e os formatos de exibição.

Esse projeto rápido leva à construção de um protótipo que será avaliado pelo cliente. O cliente por sua vez retornará um feedback à equipe de software que irá aprimorar os requisitos. A iteração vai ocorrendo conforme vamos ajustando o protótipo às necessidades dos utilizadores.

De forma geral o protótipo auxilia na identificação dos requisitos do software. Os protótipos podem ser descartados quando usados apenas para entender um determinado requisito ou pode ser utilizado como um produto evolucionário que servirá para o cliente.

⁵ Disponível <https://docplayer.com.br/5165404-Engenharia-de-software-ii.html> Obtido 05 de 03 de 20

3.2 Descrição âmbito do sistema

O aplicativo foi desenvolvido para disponibilizar uma alternativa de consulta de informações do transporte público na cidade do Mindelo e em apoio aos utilizadores deste serviço. Desenvolvido para a plataforma Android, implementando com bibliotecas de GPS e da API Google Maps para visualização de mapas.

Sendo que irá também apresentar funcionalidades que estarão disponíveis tanto em modo online, quanto em modo offline – com restrições nesta última. As principais funções disponíveis também no modo offline são: Consulta, exibição de linhas, pontos e horários.

3.3 Levantamento de requisitos

O levantamento dos dados e informações foi realizado através da recolha de dados abertos existentes na Internet do então website do operador Transcor SV, S.A. as quais se respeitou-se as exigências no que visam preservar a sua proveniência e sua abertura.

Feito o levantamento de informações, foram definidos os requisitos funcionais e não funcionais e desenhada a modelagem. Após isto, foi desenvolvida uma aplicação para dispositivos móveis, mais especificamente para a plataforma Android, para auxiliar os utilizadores do serviço de transporte coletivo de São Vicente a se deslocarem pela cidade. O principal foco da aplicação é a fácil usabilidade e melhor iteratividade por conta do utilizador.

Os requisitos de um sistema são a descrição dos serviços que ele fornecerá e as restrições que a mesma terá, levando em conta a definição do problema e da definição das ferramentas tecnológicas no desenvolvimento da solução para apoiar e resolver a situação ou situações problemáticas identificados, nesta fase se detalham os requisitos identificados durante a fase de definição do projeto, a fim de identificar novos requisitos funcionais, de informações que a solução deve satisfazer para alcançar os objetivos planeados. Os principais requisitos que o aplicativo apresentará proposta são as seguintes:

3.3.1 Requisitos funcionais

Se define as requisitos e ou se preferir-se as funções que o aplicativo será capaz de executar.

Tabela 1. Requisitos Funcionais

ID	Requisito	Descrição
RF01	Rotas	1. O sistema deverá permitir a consulta das linhas / rotas existentes;
RF02		2. O sistema deverá listar os pontos de paragens de uma rota no seu trajeto / itinerário, selecionado pelo utilizador;
RF03		3. O sistema deverá permitir listar os horários em cada ponto de paragem de uma determinada rota;
RF04		4. Mostrar no mapa o trajeto de uma rota com todos os respetivos pontos de paragem que transita;
RF05	Pontos de Paragem	5. O sistema deverá apresentar os pontos de paragens mais próximos, bem como suas distâncias em relação ao utilizador, e quais as rotas estão associadas a ela, através da localização atual do utilizador;
RF06		6. Permitir visualizar no mapa as paragens mais próximos da localização do utilizador;
RF07		7. O sistema deverá permitir listar todos os pontos de paragens existentes e opção de mostrar imagens de cada ponto;
RF08		8. O sistema deverá permitir mostrar para uma paragem selecionada, as rotas que nela transitam e respetivos horários na mesma, tendo em conta a hora atual;
RF09	Sugestão de Rotas	9. Permitir a sugestão de rotas/ linhas através do ponto de destino;
RF10		10. Pesquisar sugestão de rotas, através do ponto inicial e destino.

Fonte: Autoria própria

3.3.2 Requisitos não funcionais

Tem em conta as características que de uma certa maneira podem limitar a aplicação, sendo estas apresentadas na tabela a seguir.

Tabela 2. Requisitos não funcionais

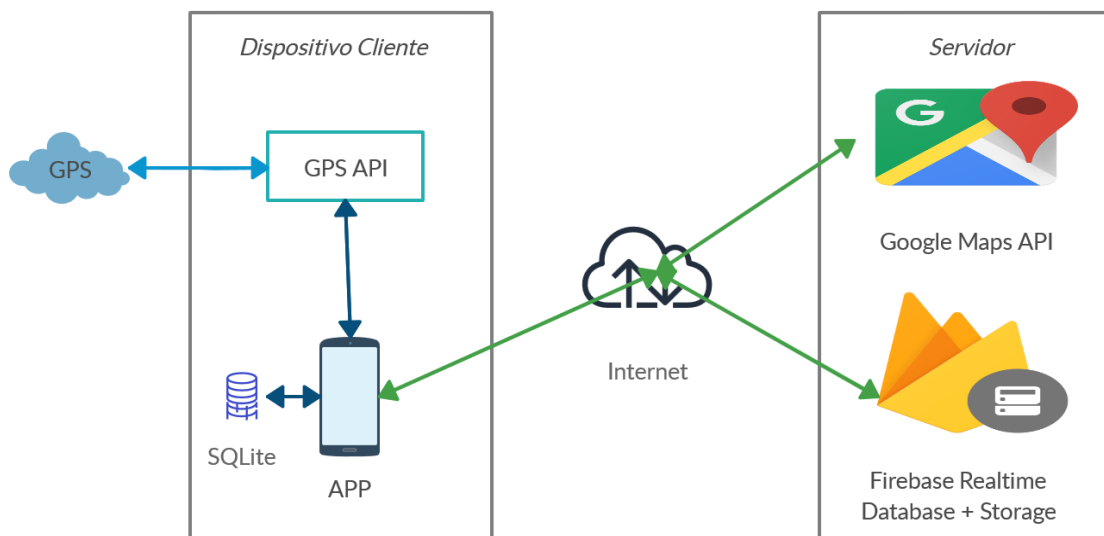
ID	Nome	Descrição
RNF01	Facilidade de uso	O sistema deverá ser fácil de usar, mostrando interfaces intuitivas, manipulação de módulos e controle de erro.
RNF02	Rendimento	Deverá garantir que o uso do aplicativo não afeta o desempenho do dispositivo, nem consumir recursos excessivos de processador, memória ou tráfego de rede.
RNF03	Disponibilidade	O aplicativo deve estar disponível para uso em qualquer momento, garantindo o uso das funcionalidades independentemente da conexão com a internet.
RNF04	Modularidade	O aplicativo será tratado através de módulos, de maneira para que cada um dos módulos lide com uma funcionalidade diferente.
RNF05	Plataforma	O aplicativo deve ser compatível com a plataforma Android versão 4.4 ou superior

Fonte: Autoria própria

3.4 Arquitetura do sistema

Para a implementação dos requisitos se observa a seguinte arquitetura:

Figura 8. Arquitetura Sistema



Fonte: Autoria própria

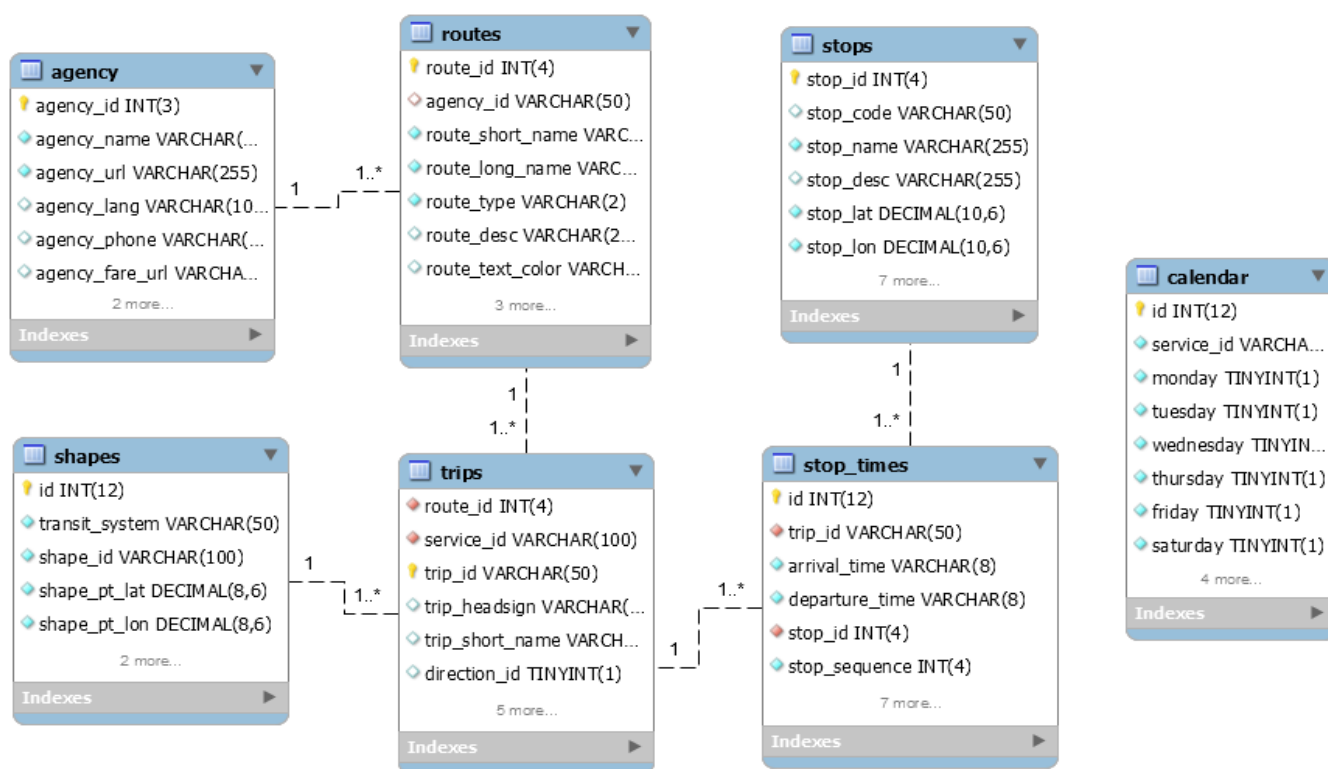
3.5 Análise e modelação do sistema

As funcionalidades do sistema desenvolvido são demonstradas através da modelagem UML, utilizada para especificar, visualizar, construir e documentar os artefactos de um sistema, geralmente com orientação a objetos.

3.5.1 Modelo Entidade – Relacionamento (ER)

A base de dados foi desenvolvida tendo por referência a estrutura baseada no modelo estático da GTFS – *Google Transit Feed Specification* ⁶, a qual é usado no Google Transit, nomeadamente para pesquisa de transportes no Google Maps. Estando em uso neste projeto, uma versão simplificada e adaptado do mesmo, na qual se apresenta na figura a seguir.

Figura 9. Modelo Entidade Relacionamento baseado no modelo GTFS



Fonte: Autoria própria

A função de cada entidade presentes na figura 9 está descrita a seguir:

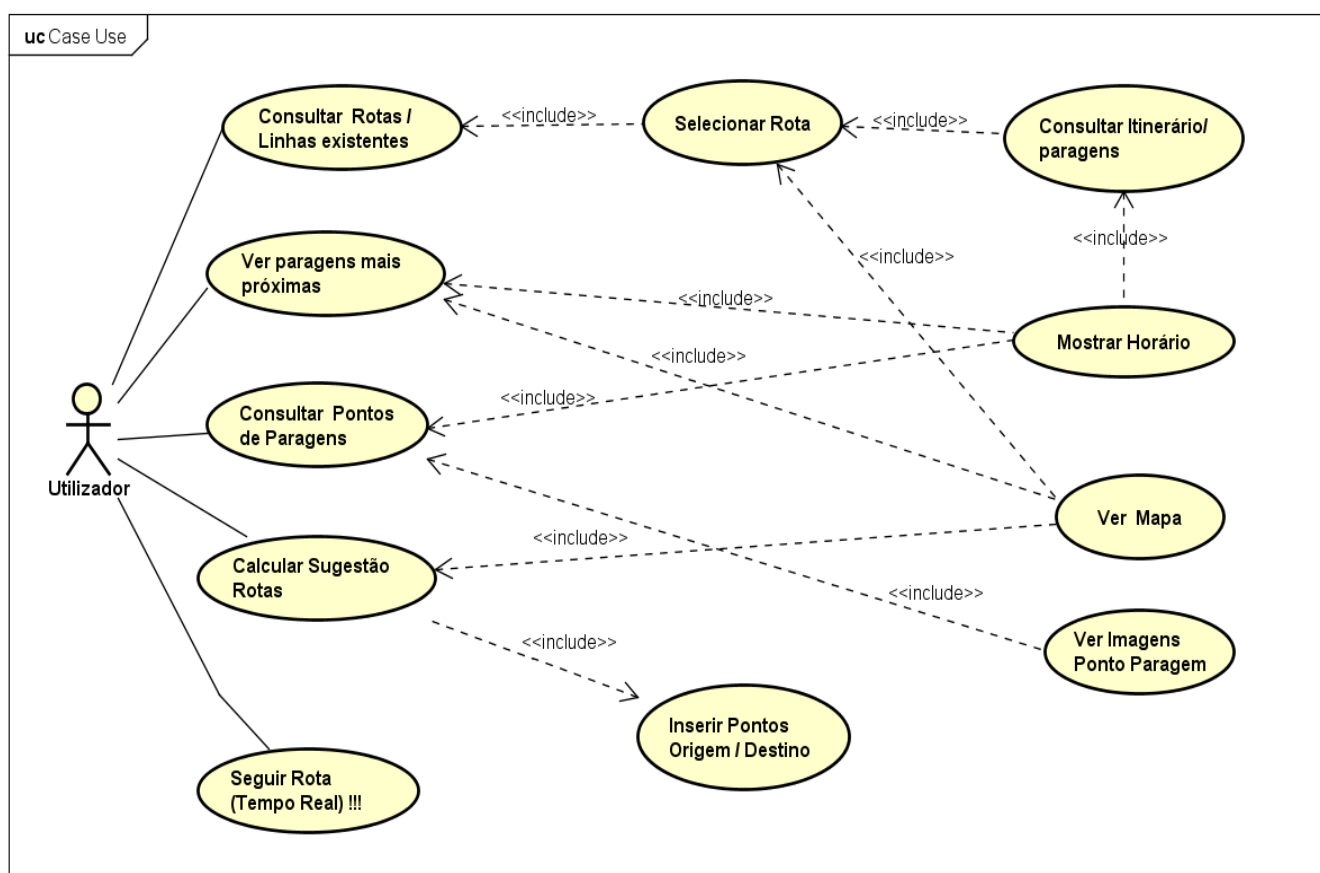
⁶ Disponível em <https://developers.google.com/transit/gtfs>

- a) **agency**: responsável por armazenar os dados dos operadores do serviço de transporte;
Ex: Transcor SA, Amizade, e neste projeto tem-se por referência a Transcor.
- b) **routes**: responsável por armazenar as rotas existentes para uma ou vários operadores;
Ex. A Transcor e suas rotas, designadas de Linhas L1, L2,..., L11, perfazendo um total de 11 rotas / linhas, excetuando as especiais.
- c) **trips**: responsável por armazenar as designadas “voltas”, de cada uma das rotas ao longo do dia e ou ao longo da semana.
Ex. Partindo do pressuposto de que as rotas são circulares, isto é, onde a rota se inicia é o mesmo onde ela termina, temos que a cada uma volta completa corresponde a uma trip. Para efeitos de exemplificação considera-se que a rota / linha nº5 ao longo do dia efetua 100 trips (voltas). Caso a rota não se apresenta a forma circular, então separa as trips em direções por exemplo, inbound (ida) e outbound (volta).
- d) **stop_times**: responsável por armazenar os horários para cada ponto de paragem das rotas ao longo das suas trips (voltas), ao longo do dia ou semana (calendar);
Ex. Se a rota / linha nº 5 (route), efetua 100 voltas (trips) ao longo do dia (calendar), e tem 20 pontos de paragem (stops), significa que para o ponto de paragem nº1 (stop), vamos ter 100 registos de horários (times).
- e) **stops**: responsável por armazenar os nomes dos pontos de paragens e as suas respetivas coordenadas geográficas / localização.
Ex. Ponto de paragem (stop): xpto, latitude:0.00000, longitude:0.00000
- f) **shapes**: responsável por armazenar o percurso / itinerário de cada linha sob forma de pontos de localização (lat,lon) para fazer a “grafia” da mesma, por exemplo no maps.
- g) **calendar**: responsável por armazenar os dias em que a rota circula, caso este não circular todos os dias, ou tem uma frequência diferente para as mesmas;

3.5.2 Diagrama de Casos de Uso

Os Casos de Uso descrevem o comportamento esperado do produto em uma hipotética interação com o utilizador. Podendo ser expressos através de diagramas e fluxos de casos de uso. Os diagramas de casos de uso descrevem os relacionamentos entre os casos de uso entre si e com os atores, enquanto os fluxos descrevem com detalhes cada caso de uso em sequência há a descrição do caso de uso, em que é explicado o fluxo principal da aplicação, para melhor entendimento dos eventos.

Figura 10. Diagrama de Casos de Uso



Fonte: Autoria própria

Em sequência se faz uma descrição de um caso de uso em que se exemplifica o fluxo principal da aplicação, para melhor entendimento dos eventos, como se demonstra na tabela a seguir:

Tabela 3. Fluxo de Eventos de Caso de Uso

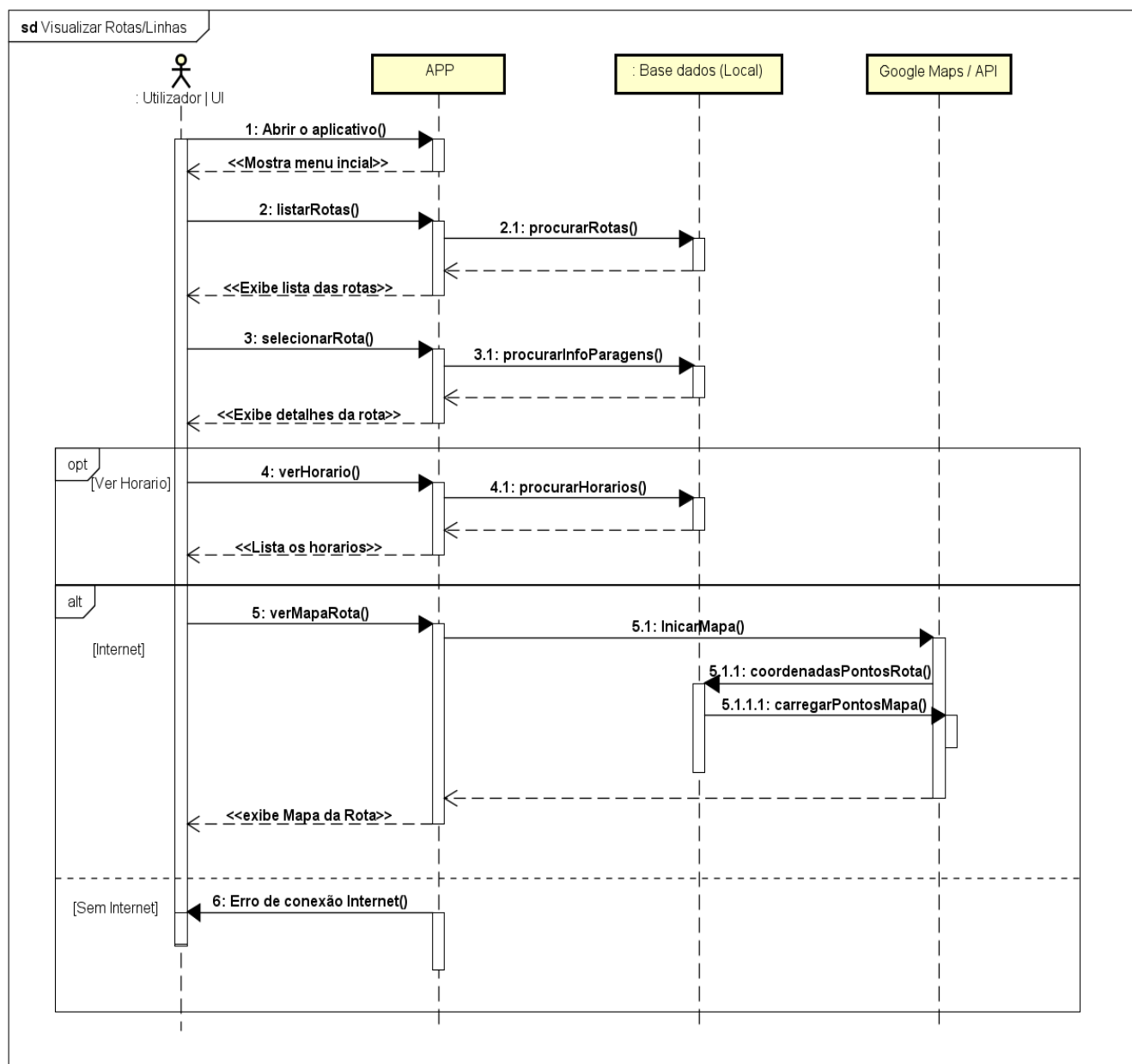
Caso de uso	Selecionar uma rota e obter suas informações.
Descrição	O utilizador deseja selecionar uma rota/linha de transporte para poder então visualizar informações sobre as paragens, horários e ver o mapa da rota.
Ator	Utilizador
Pré-condição	<ul style="list-style-type: none"> • O utilizador deve ter selecionado a opção do menu inicial “Rotas”, a qual lista todas as existentes do sistema; • O Sistema tem guardado e tem conhecimento das rotas, pontos de paragens, respetivos horários e dados para carregar Mapa.
Pós-condição	O utilizador encontra e seleciona uma rota, podendo então visualizar informações detalhadas sobre a mesma.
Fluxo Eventos Principal	<ol style="list-style-type: none"> 1. O utilizador informa ao sistema que deseja selecionar uma rota; 2. O sistema exibe as rotas / linhas que tem registado; 3. O utilizador seleciona uma rota / linha, sendo então capaz de visualizar os pontos pelos quais essa rota passa, seus horários gerais, bem como poder ver qual o percurso dessa rota via mapa; 4. Após navegar nas informações disponíveis desejadas, o utilizador finaliza a operação.
Fluxo Alternativo	<p>* A qualquer momento, o utilizador cancela a operação.</p> <p>3a. O utilizador seleciona uma rota, visualiza os pontos pela qual passa, seleciona a opção ver Mapa, seleciona um ponto nesse mapa e visualiza os horários de paragens neste ponto de todas as rotas que transitam nela e próximos horários das rotas nesse ponto.</p>
Restrições	O Mapa só será carregado se estiver com conexão a Internet.
Casos de uso complementar	Consultar Itinerário / paragens, Mostrar Horário, Ver Mapa.

Fonte: Autoria própria.

3.5.3 Diagramas de Sequência

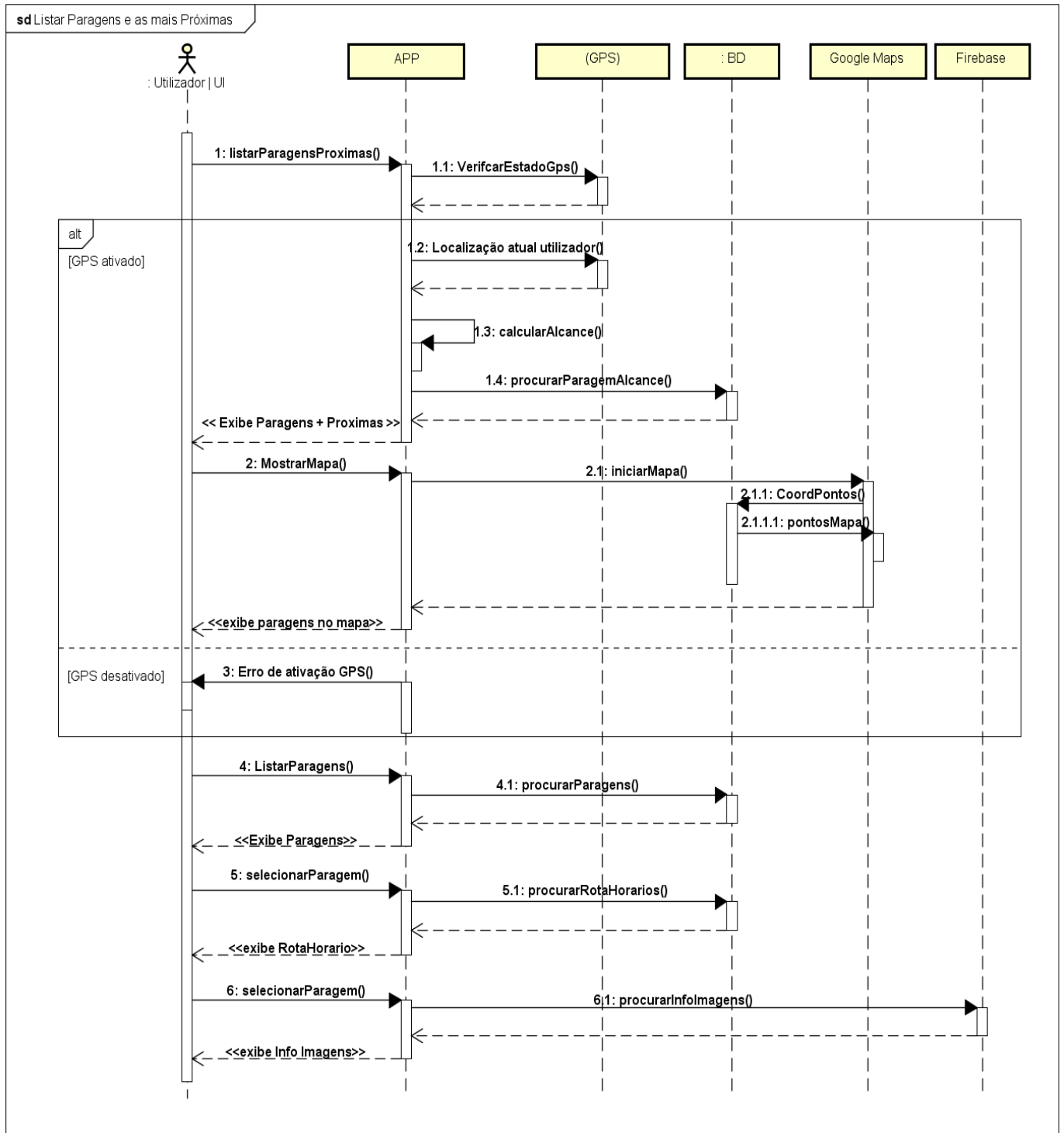
O diagrama de sequência representa as interações dos atores (utilizadores, aplicativo base de dados e servidor externo), com envio de mensagens de solicitação e respostas de retorno entre eles, em uma linha de tempo para acompanhar o processo, conforme figura.

Figura 11. Diagrama de Sequencia Rotas



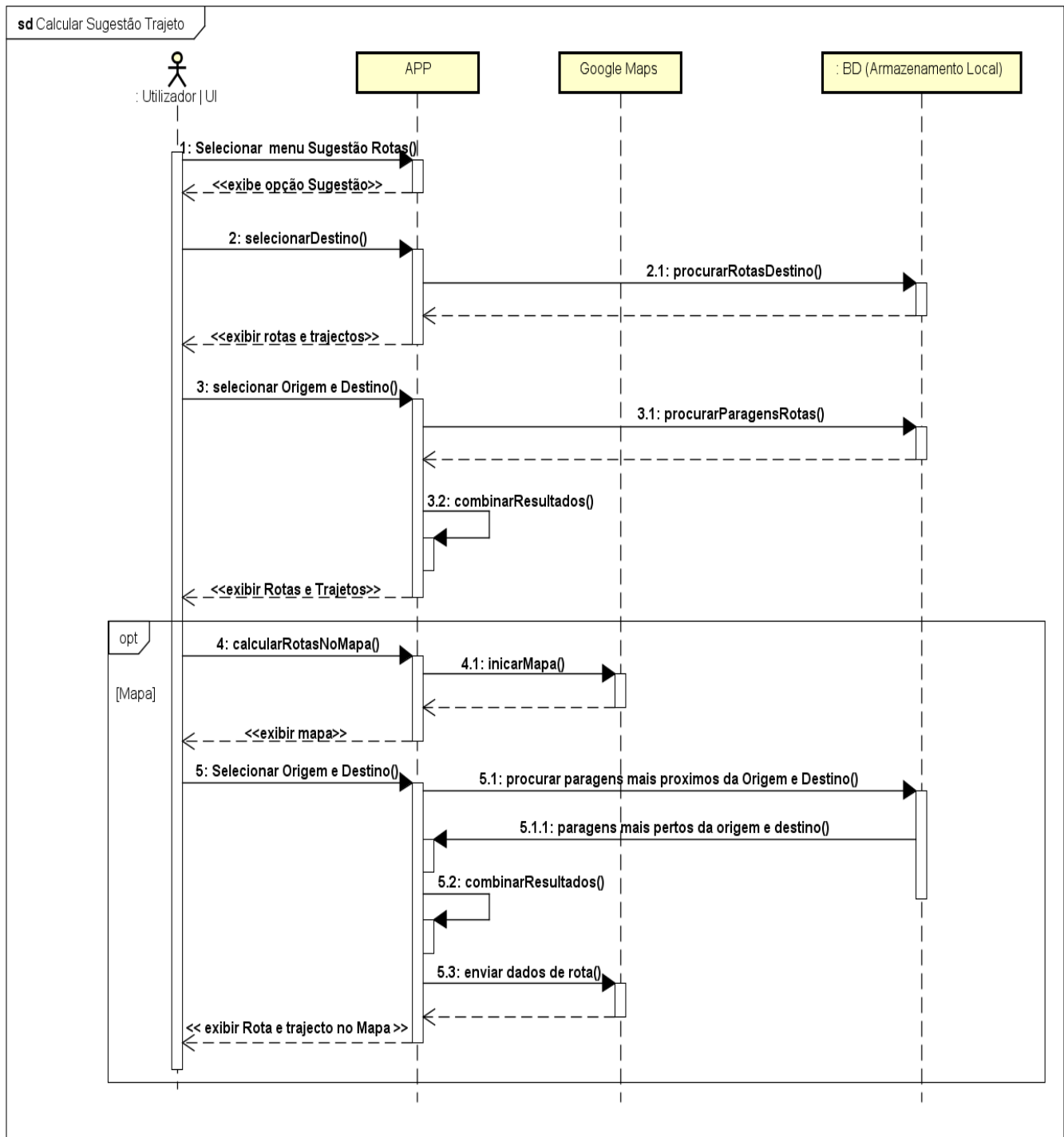
Fonte: Autoria própria

Figura 12. Diagrama de Sequencia Pontos de Paragem



Fonte: Autoria própria

Figura 13. Calcular Sugestão Trajetos

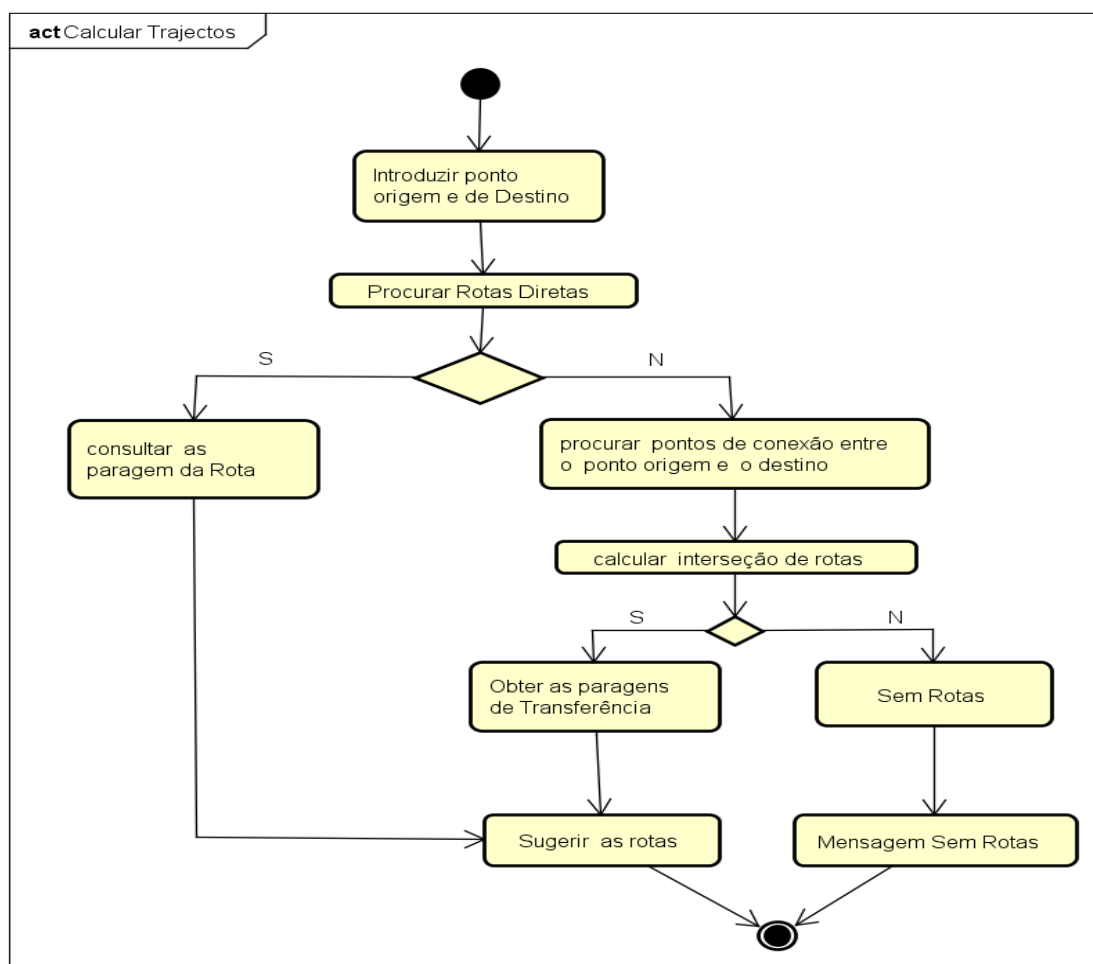


Fonte: Autoria própria

3.5.4 Diagrama de Atividade

O diagrama de atividade representa um diagrama comportamental, através da qual podemos modelar partes do comportamento de um software, ilustrando graficamente como será o funcionamento do software, como será a execução de alguma de suas partes, como será a atuação do sistema na realidade, na qual ele está inserido.

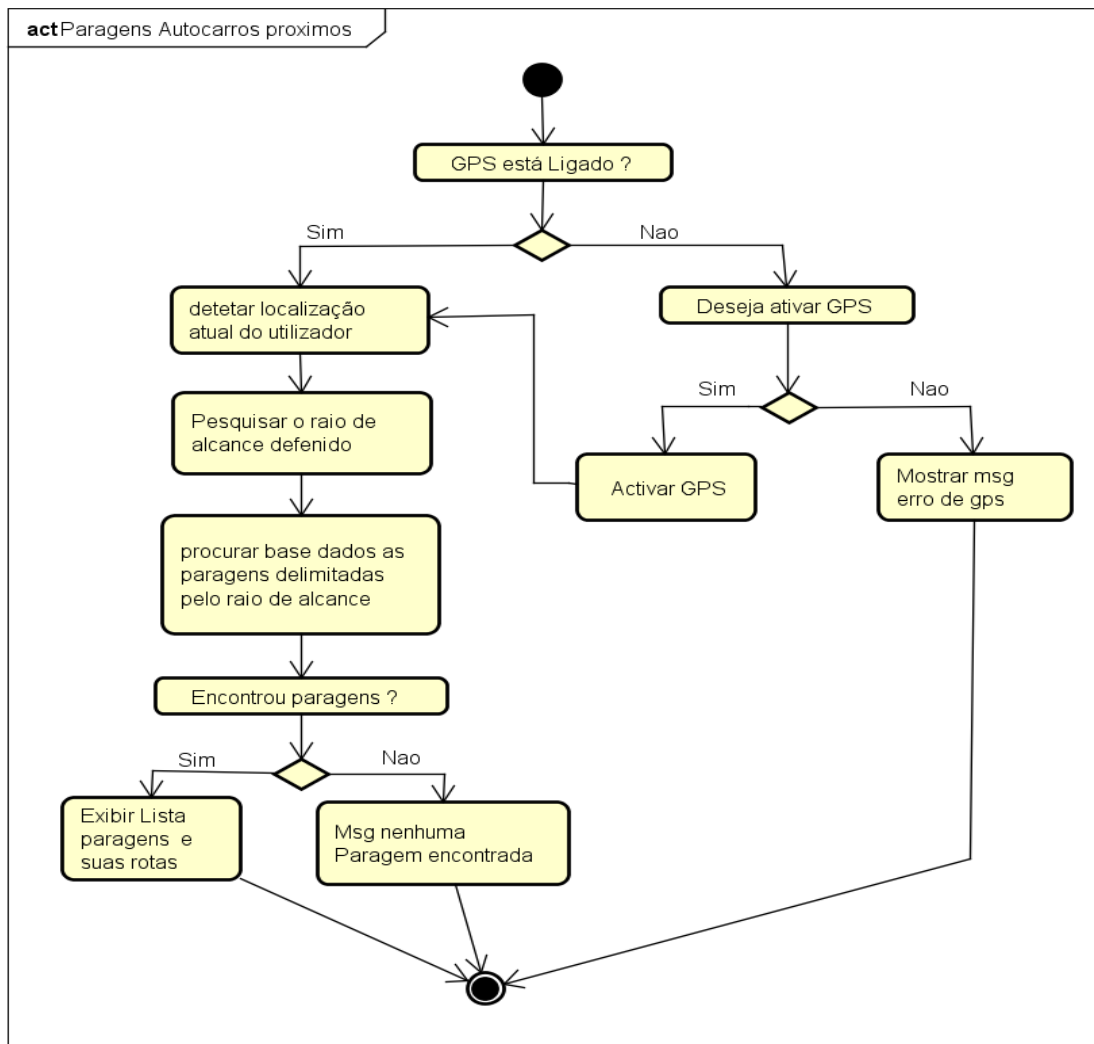
Figura 14. Diagrama Atividade Calcular Trajeto



Fonte: Autoria própria

Este Diagrama representa o comportamento, do calcular sugestão de rotas, sem recorrer ao uso de mapa, na qual se procura por sugestões diretas de rotas que consiste em verificar se a origem e destino selecionados podem ser encontrados numa mesma rota, caso não encontre essas duas referências numa mesma rota, procura-se fazer uma intersecção para encontrar rotas indiretas, que implicam a interseção de duas rotas.

Figura 15. Diagrama de atividade encontrar paragens mais próximos do utilizador



Fonte: Autoria própria

Este Diagrama representa o comportamento, de procurar os pontos de paragens mais pertos do utilizador, sem recorrer ao uso de mapa, na qual esta só será possível se o utilizador estiver com a localização ativada ou der permitir ao aplicativo para ativar a funcionalidade de localização.

3.5.5 Diagrama de Classes

São apresentadas as classes utilizadas para o desenvolvimento do aplicativo, a estrutura das classes, com seus atributos e métodos e a relação entre as classes, conforme pode ser vista na Apêndice A.

CAPITULO IV

4 DESENVOLVIMENTO DO SISTEMA – PROJETO

A utilização de smartphones tem-se tornado cada vez mais comum, principalmente para acesso à Internet e busca de informações. Tecnologias para desenvolvimento de aplicações mobile também tem-se popularizado entre os desenvolvedores, principalmente para o ambiente Android, considerando a grande popularização dessa plataforma entre os utilizadores.

Para o desenvolvimento mobile, (Ambros, 2013) especificou três tipos de aplicações: nativas, web e híbridas. Esse autor ressalta que a grande vantagem do desenvolvimento nativo é poder utilizar todas as funcionalidades presentes nos dispositivos, como: camera, *Global Positioning System* (GPS), sistema de notificação nativo e funcionar sem conexão com a Internet.

Considerando esse contexto, este trabalho visa desenvolver uma aplicação mobile nativa que auxilie os utilizadores a encontrarem informações sobre autocarros tendo como uma das bases exatamente a localização do utilizador, sendo assim torna recomendável recorrer ao desenvolvimento nativo para acesso a essas funcionalidades.

Sendo assim neste capítulo, se apresenta todos as tecnologias e ferramentas utilizados para a realização deste trabalho, bem como o desenvolvimento do aplicativo proposto e suas demonstração das interfaces das suas funcionalidades e tópicos relevantes das mesmas.

4.1 Tecnologias e Ferramentas de Desenvolvimento

Para este projeto as principais Tecnologias e ferramentas de desenvolvimento foram as que se encontram detalhadas a seguir.

Tabela 4- Ferramentas e Tecnologias Desenvolvimento

Ferramenta /Tecnologia	Versão	Referência	Finalidade
Android Studio	3.6.1	https://developer.android.com/studio	Ambiente de Desenvolvimento (IDE).
Java	1.8.0	https://www.java.com	Linguagem de Programação.
SQLite	3.0	https://www.sqlite.org	Base de Dados.
Firebase	16.0.0	https://firebase.google.com	Base de Dados NoSQL.
Google Maps Platform API	v2	https://developers.google.com/maps/documentation	Serviço de visualização de mapas.
Balsamiq Mockups	3.5.17	https://balsamiq.com/	Desenhos de interface utilizador para protótipo.
Astah Profissional	8.0.0	https://astah.net/	Documentação da Modelagem baseada na UML.
PHP, JS, HTML	---	---	Desenvolvimento Web.

Fonte: Autoria própria

A seguir é apresentado um resumo sobre as principais ferramentas utilizadas.

4.1.1 Android

Figura 16. Logotipo do Android



Fonte: Android⁷

Segundo (Glauber, 2015), o Android é um conjunto de softwares para dispositivos móveis que inclui um sistema operacional, um middleware e aplicações chave. O sistema operacional tem como base do kernel do Linux, que é responsável pelo gestão de processos,

⁷ Disponível <https://developer.android.com/guide/platform?hl=pt-br> Obtido em 05 de 03 de 2020

drivers, memória e energia. O middleware controla a interação dos aplicativos instalados com o aparelho e as aplicações-chave são programas comuns, como discador, navegador, etc.

Nos dias atuais o sistema Android é usado em diversos e mais variados dispositivos como TVs, tablets wearables, seu sistema é distribuído pela Apache sob licença de mesmo nome, permitindo uma personalização sem precisar compartilhar as mudanças com os seus concorrentes, porém para ser homologado precisa passar por vários testes para evitar que suas mudanças não afetem as APIs (*Application Programming Interface*).

O Android vem passando por diversas atualizações, a primeira versão saiu em 2008 e se atribui um nome de doce e um número sequencial que é chamado de API Level. Isso é extremamente importante para saber os recursos disponíveis entre bibliotecas e classes que cada versão disponibiliza, podendo assim criar a aplicação que suporta a versão desejada.

4.1.2 Android Studio

Figura 17- Logotipo Android Studio



Fonte: Android Studio ⁸

Android Studio é um ambiente de desenvolvimento disponibilizado pelo Google para o sistema Android. Essa ferramenta vem para fornecer novas maneiras de desenvolvimento de aplicativos mais modernos e práticas.

Na ferramenta, praticamente toda a estrutura do projeto fica dentro do diretório SDK, que é um kit de desenvolvimento de software, um pacote de programação que permite a um

⁸ Disponível <https://developer.android.com> Obtido em 05 de 03 de 2020

programador desenvolver aplicações para uma plataforma específica. Normalmente, um SDK inclui uma ou mais APIs, ferramentas de programação e documentação. Conta juntamente com o SDK um sistema de gestão baseado em Gradle onde projeto criado no Android Studio já vem por padrão, sendo assim, temos um arquivo de configuração para o projeto principal e um para cada módulo.

O Android Studio tem funcionalidades bem práticas, como poder visualizar toda alteração em tempo real, e ver como aparecerá em diferentes dispositivos Android com diferentes resoluções. O programa também utiliza de um sistema de clicar e arrastar componentes pela interface do utilizador, um método comum entre outros softwares, mas que fez toda diferença se tratando de praticidade.

4.1.3 Linguagem de Programação – JAVA

Figura 18. Logotipo Java



*Fonte: Java*⁹

Para qualquer iniciante e ou desenvolvedor de aplicativos mobile em todo o mundo, uma das primeira e mais preferida linguagem de programação para um aplicativo Android é o Java, uma das razões pelas quais é simplesmente é uma das linguagens de programação oficial para desenvolvimento de aplicativos Android e por ser bem documentada e com grande suporte e comunidades.

O Java foi inicialmente desenvolvido pela Sun Microsystems em 1995 e vendida para a Oracle, e é usado para uma ampla gama de softwares. O código Java é executado por uma máquina virtual, que é executada em dispositivos Android e interpreta o código, sendo uma linguagem bem documentada, o que permite um aprendizagem rápida.

⁹ Disponível <https://www.java.com> Obtido em 05 de 03 de 2020

4.1.4 SQLite

Figura 19. Logotipo SQLite



Fonte: SQLite ¹⁰

O SQLite segundo o seu website, foi iniciado no ano 2000 e se baseia em uma biblioteca que implementa um base de dados transacional, autocontido, e um mecanismo de base de dados que se incorpora no SQL, diferente dos outros tipos de base de dados ele não usa servidor separado, ele lê e grava suas informações diretamente em discos comuns sem necessidade de servidor e nem de configurações.

Pode se dizer que é uma base de dados completo com índices, disparos, múltiplas tabelas e exibições com vários comandos SQL compatíveis.

O SQLite é uma biblioteca muito compacta, podendo ser menor que 600kb na maioria dos casos dependendo da plataforma de destino, ele se executa mais rápido quando se dá mais memória no dispositivo. No entanto seu desempenho na maioria das vezes é excelente por ser leve, compacto e rápido, e o código do SQLite tem domínio público e gratuito, podendo sofrer alterações para serem adaptadas aos desenvolvedores.

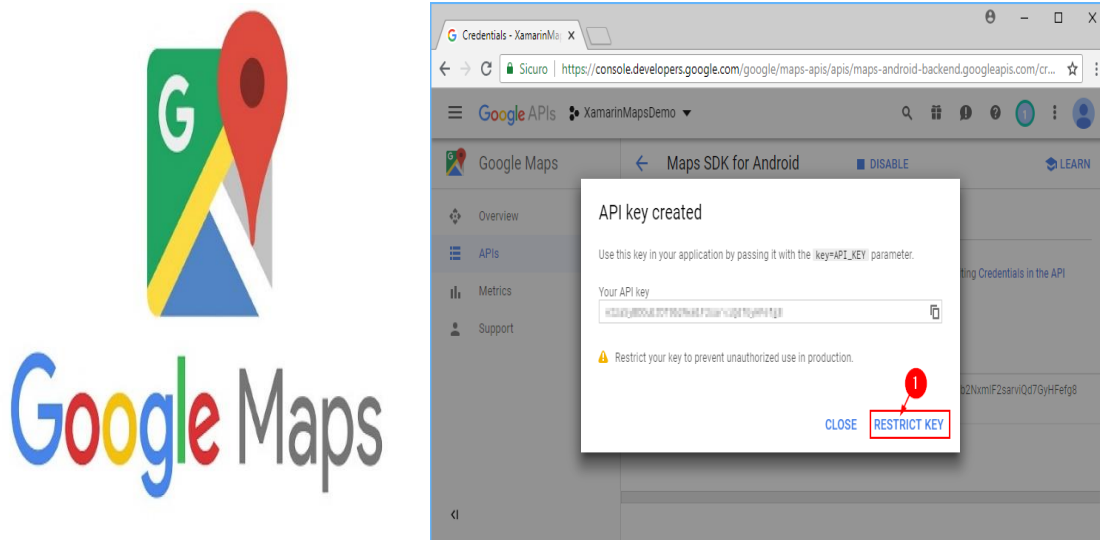
De realçar as bibliotecas, ou libraries que fazem parte da infraestrutura do Android, e que consistem em um conjunto de bibliotecas C/C++ usadas por diversos componentes do Android e possuem funções específicas. De entre essas bibliotecas, destacam-se:- SQLite: implementação de uma base de dados relacional disponível para todos os aplicativos, o que garante que é uma biblioteca suportada nativamente pelo Android.

O SQLite é um software livre, de domínio público e multiplataforma, não necessita de instalação, configuração ou manutenção.

¹⁰ Disponível <https://www.sqlite.org/about.html> Obtido em 05 de 03 de 2020

4.1.5 Google Maps Platform Android API

Figura 20. Google Maps



Fonte: Google Maps ¹¹

O Google disponibiliza uma API de mapas, chamada Google Maps Android API V2. Toda a parte visual da API foi desenvolvida utilizando diversas novas formas de visualização do mapa em 2D e 3D, obtendo muito mais desempenho nas interações e animações da API. Uma novidade é a visualização 3D ser feita automaticamente se o zoom estiver sendo utilizado podendo visualizar imagens caso na cidade tenha registado imagens 3D.

Para utilizar do recurso de visualização de mapas com e ou sem recurso a GPS é necessário o uso da API do Google de Mapas, aonde se obtém permissão de acesso ao maps e todos os recursos de visualização de localização. Para que a API funciona é necessário fazer o registo de uma conta como desenvolvedor da Google para solicitar e obter uma chave de autenticação que nos dá uma licença de uso do mapa da Google, sendo que para solicitar a chave é preciso fornecer o código do certificado digital utilizado para assinar sua aplicação do seu projeto Android.

Porém essa chave tem um limite de acesso para ser gratuita, para que se aumente esse recurso é necessário fazer registo de uma conta com pagamento para se obter um maior limite de acesso para um número de utilizações e solicitações por dia.

¹¹ Disponível <https://developers.google.com/maps> Obtido em 05 de 03 de 2020

4.1.6 Firebase

Figura 21. Firebase



Fonte: Firebase ¹²

O **Firebase** é um **BaaS** (*Backend as a Service*) para aplicações Web e Mobile do Google, foi lançado em 2004 e com crescimento de uso ao longo dos tempos, se tornando uma ferramenta que hoje para alguns projetos é a melhor opção, devido a quantidade de serviços oferecidos por ele, além da facilidade de implementação.

O Firebase oferece uma gama de serviços que podem ser utilizados, entre eles: Analytics, Realtime Database, Authentication, Cloud Messaging, Storage, AdMob, entre muitos outros.

Neste projeto o seu uso se dá pela utilização dos seguintes serviços:

Realtime Database

Uma Base de Dados em Tempo Real que permite o armazenamento e sincronismo dos dados entre utilizadores e dispositivos em tempo real com uma base de dados NoSQL hospedado na nuvem.

Os dados atualizados são sincronizados em todos os dispositivos conectados em segundos. Além disso, seus dados permanecem disponíveis caso seu aplicativo fique offline, o que oferece uma ótima experiência do utilizador, independentemente da conectividade de rede.

¹² Disponível <https://firebase.google.com> Obtido em 05 de 03 de 2020

Storage

O Firebase Cloud Storage permite que se armazene e compartilhe facilmente imagens, áudio, vídeo ou outro conteúdo gerado pelos utilizadores, usando um armazenamento de objetos poderoso, simples e econômico criado para a escala do Google.

A escolha deve-se a entre outros por trazer benefícios de desenvolvimento em:

- **Pequenas Aplicações:** ideia para um projeto, mas para a qual não se tem tempo para desenvolver o back-end ou não se tem conhecimentos suficientes para tal.
- **Protótipos:** Quando se precisa criar uma aplicação demonstração, ou um projeto universitário ou então apenas para testar as requisições de um aplicativo e não é necessário criar todo o back-end, utilizando o Firebase para agilização do processo.

4.1.7 Astah

Figura 22. Logotipo Astah



Fonte: Astah ¹³

Astah Community é um software para modelagem UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada) com suporte a UML 2, desenvolvido pela Change Vision, Inc. Anteriormente conhecido por JUDE, um acrônimo de *Java and UML Developers Environment* (Ambiente para Desenvolvedores UML e Java), disponibiliza para desenvolvimento, os diagramas de Classes, Casos de Uso, Sequência, Comunicação, Máquina de Estados, Atividade, Componentes, Implantação e Diagrama de Estrutura.

¹³ Disponível <https://astah.net> Obtido em 05 de 03 de 2020

4.2 Equipamentos

A Tabela 5 apresenta todos os equipamentos requeridos para o desenvolvimento do projeto e os requisitos a serem atendidos.

Tabela 5. Equipamentos necessários para desenvolver Aplicativo

Equipamento	Requisitos
Computador	Sistema Operativo Windows 10 64 Bits SSD 256 GB / 6 GB de RAM
Dispositivo móvel	Smartphone, Tablet, S.O Android versão 4.4 ou superior

Fonte: Autoria própria

4.3 Definição de Protótipo a se Obter

Para este projeto, tendo em conta a abordagem dos componentes, ferramentas e técnicas necessárias para implementar o aplicativo móvel, pretende-se com a mesma verificar se cumpre com as seguintes funcionalidades:

- Visualizar as rotas de autocarros existentes;
- Traçar a rota que o autocarro segue em um mapa;
- Mostrar as informações das rotas de autocarro que passam por um ponto de paragem;
- Encontrar rotas através da introdução do ponto de destino;
- Mostrar informações detalhadas de uma rota selecionada;
- Obter informações de um ponto de paragem selecionada;
- Mostrar pontos de paragens mais próximos ao utilizador;
- Realizar sugestões de autocarros que um utilizador pode seguir para chegar de um ponto a outro na cidade do Mindelo.

4.4 Apresentação do Sistema

São apresentadas a seguir, as telas do aplicativo para consulta de informações de autocarro do transporte coletivo. A tela principal, apresentada na figura 23, possui uma lista em que o utilizador pode escolher as funcionalidades que o utilizador quer pesquisar.

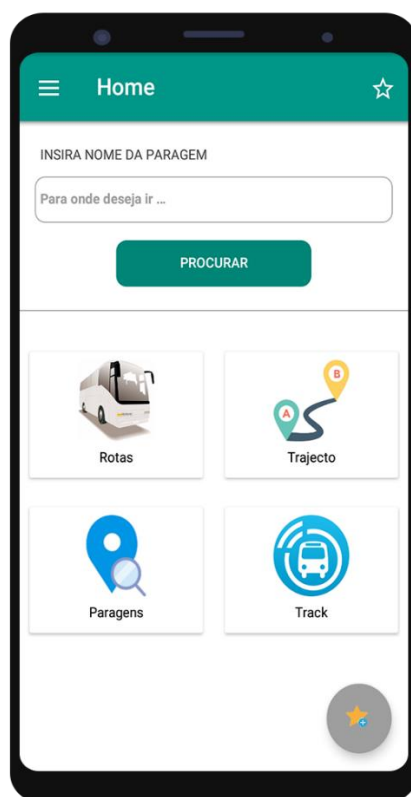
4.5 Implementação do Sistema - Protótipo

De seguida se apresentará as interfaces de utilizador sobre a forma de prints e códigos que suportam para a sua implementação, da qual se utilizou a linguagem programação Java.

4.5.1 Prints de Tela e codificação

4.5.1.1 Menu Principal

Figura 23. Página Inicial Protótipo

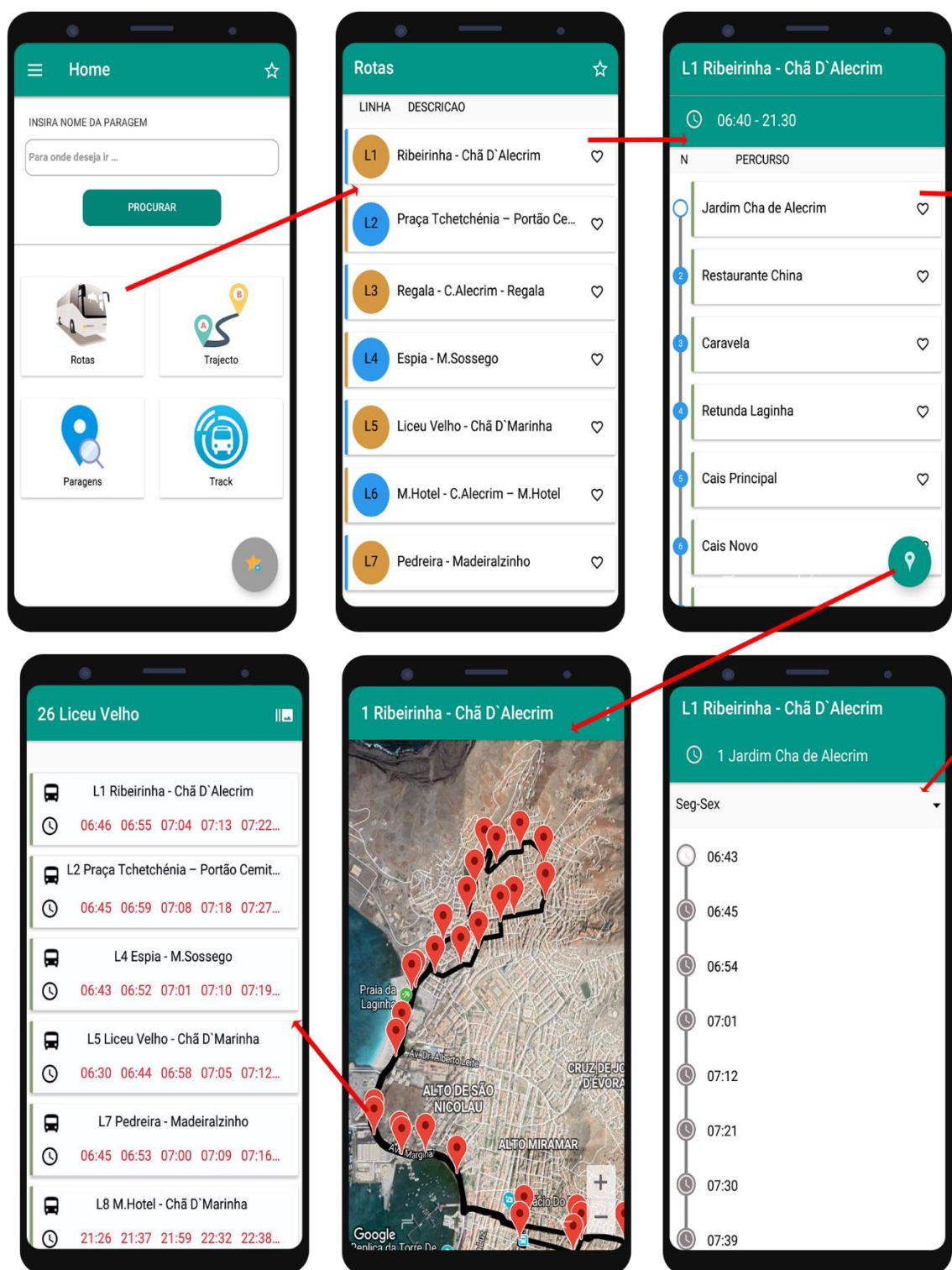


Fonte: Autoria própria

Esta tela apresenta a Dashboard principal do aplicativo, onde se disponibiliza as principais funcionalidades para o utilizador.

4.5.1.2 Rotas

Figura 24. Informações detalhadas das Rotas



Fonte: Autoria própria

Permite que sejam obtidos todas as informações das rotas e detalhes da mesma, e como mostra a figura 24 acima exposto, as telas representam quase uma sequência da tela anterior, na medida que apresenta informações complementares, de uma tela selecionada anteriormente.

Dentro delas a destacar em termos de codificação a tela 5 da figura acima exposto, em que se representa as rotas via Google maps, com os pontos de paragens e um designado “polyline” que representa o trajeto completo da rota, que em termos de codificação poderá ser parcialmente verificada na figura 25 abaixo exposto.

Figura 25. Código parcial visualizar rotas no Mapa

```

/* carregar Google Maps */
@Override
public void onMapReady(GoogleMap maps) {
    mapi = maps;
    MapUtils.addStopMarkersByRoute(dbHelper, stopIds, mapi);
    MapUtils.addPolylineMarkersByRoute(dbHelper, routeNo, mapi);
    maps.setMapType(GoogleMap.MAP_TYPE_HYBRID);
    maps.animateCamera(CameraUpdateFactory.zoomTo(14.0f));
    maps.setOnMarkerClickListener(this);
    maps.getUiSettings().setZoomControlsEnabled(true);
}

/* */
public static void addStopMarkersByRoute(DBHelper dbHelper, String stopIds[], GoogleMap map) {
    // Adds markers to the map corresponding to the location of the stops
    Cursor stops = dbHelper.queryStopInfoForMap(stopIds);
    int noStops = stops.getCount();
    Double stopLatitude;
    Double stopLongitude;

    stops.moveToPosition(0);
    for (int i = 0; i < noStops; i++) { // TODO: implement more efficient iteration
        stopLatitude = Double.parseDouble(stops.getString(2));
        stopLongitude = Double.parseDouble(stops.getString(3));
        /* (...) */
    }

    public static void addPolylineMarkersByRoute(DBHelper dbHelper, String routeNo, GoogleMap map) {
        // Adds markers to the map corresponding to the location of the stops
        // query para buscar As Coordenadas para desenhar polyline

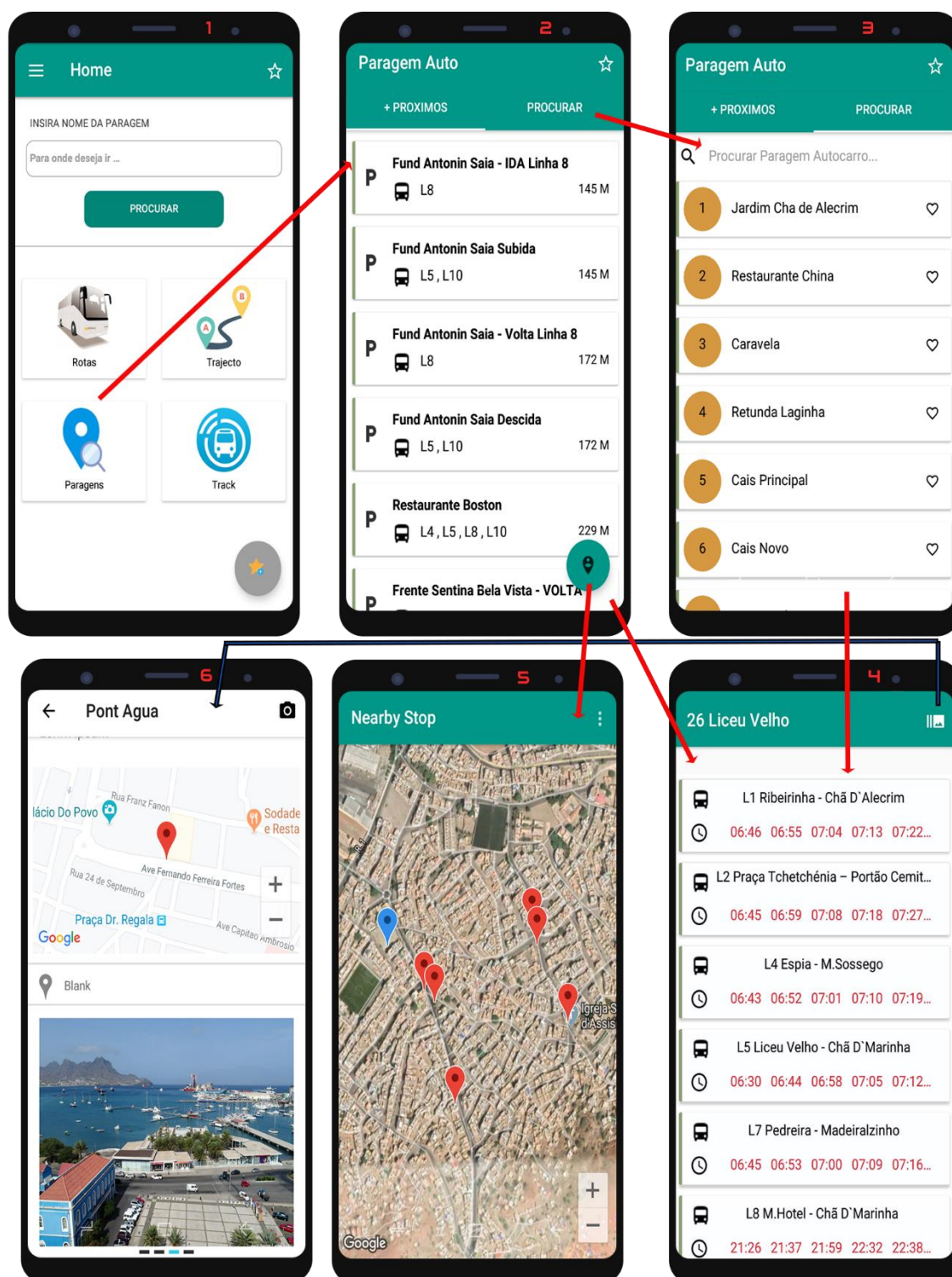
        List<RouteMap> polylatlon = dbHelper.getRoutePolyline(routeNo);
        /* (...) */
        if(polyline == null)
        {
            int index = 0; // whatever index
            //colocar o polyline inside
            PolylineOptions polylineOptions = new PolylineOptions();
            // Create polyline options with existing LatLng ArrayList
            polylineOptions.addAll(latlonx);
            polylineOptions.width(10).color(Color.BLACK).geodesic(true);
            map.addPolyline(polylineOptions);
            ///
        }else{
            polyline.setPoints(latlonx);
        }
    }
}

```

Fonte: Autoria própria

4.5.1.3 Pontos de Paragens

Figura 26. Informações detalhadas Pontos de Paragens



Fonte: Autoria própria

Pontos de paragens mais próximos da localização do utilizador via API GPS

Permite que seja possível identificar os pontos de paragens mais perto da localização atual do utilizador, desde que a opção de GPS esteja ativado, ou então seja dado permissão ao aplicativo, para que este ative o GPS, aplicando posteriormente uma fórmula, que dentro de um raio de procura predefinida, se consiga procurar e mostrar esses pontos caso estes existam, este representado na tela 2 da figura 26, acima exposto.

Em termos de codificação da mesma, apresenta-se na figura 28 abaixo exposto, código de fórmula aplicada para se conseguir os resultados da busca pelas paragens.

Pontos de paragens mais próximos da localização do utilizador via Google Maps

Permite que via uso do Google maps e sua API de Mapas, mostrar um mapa com designado “markers” que designam neste caso os pontos mais próximos, permitindo ao se mover a camera, atualizar esses mesmos markers com novos pontos de paragens mais próximos, este representado na tela 5 da figura 26 acima exposto e que em termos de codificação parcial representada na figura 29 abaixo exposto.

Figura 27. Exemplo código para se obter a posição atual do utilizador / dispositivo

```
private void getDeviceLocation() {  
    /*  
     * Get the best and most recent location of the device, which may be null in rare  
     * cases when a location is not available.  
     */  
    try {  
        // if (mLocationPermissionGranted) {  
        Task<Location> locationResult = mFusedLocationProviderClient.getLastLocation();  
        locationResult.addOnCompleteListener(this, new OnCompleteListener<Location>() {  
            @Override  
            public void onComplete(@NonNull Task<Location> task) {  
                // if (task.isSuccessful()) {  
                if (task.isSuccessful() && task.getResult() != null) {  
                    // Set the map's camera position to the current location of the device.  
                    mLastKnownLocation = task.getResult();  
                    if (mLastKnownLocation != null) {  
                        getUser = new LatLng(mLastKnownLocation.getLatitude(), mLastKnownLocation.getLongitude());  
                    } else {  
                        getUser = new LatLng(SV_LAT, SV_LONG);  
                        Toast.makeText(activityContext, "OP2", Toast.LENGTH_SHORT).show();  
                    }  
                    mapi.addMarker(new MarkerOptions().flat(true).position(new LatLng(mLastKnownLocation.getLatitude(),  
                        mLastKnownLocation.getLongitude()))  
                        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE)).title("You").snippet("0"));  
                } else {  
                    Log.d(TAG, "Current location is null. Using defaults.");  
                    Log.e(TAG, "Exception: %s", task.getException());  
                    // mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(mDefaultLocation, DEFAULT_ZOOM));  
                    mMap.getUiSettings().setMyLocationButtonEnabled(false);  
                }  
            }  
        });  
    } catch (SecurityException e) {  
        Log.e("Exception: %s", e.getMessage());  
    }  
}
```

Fonte: Autoria própria

Figura 28. Código parcial de fórmula de procurar paragens próximas via GPS

```

/* implementacao de procura de pontos de paragens
baeados em formulas e calculos tendo em conta o raio defenido. */

NearbyBusStopsDbTask(float currentLat, float currentLong, float radius)
{
    this.currentLat = currentLat;
    this.currentLong = currentLong;
    this.radius = radius;
}

/**
 * Calculates the end-point from a given source at a given range (meters)
 * and bearing (degrees). This methods uses simple geometry equations to calculate the end-point. */
private static PointF calculateDerivedPosition(PointF point, double range, double bearing)
{
    double EarthRadius = 6371000; // m

    double latA = Math.toRadians(point.x);
    double lonA = Math.toRadians(point.y);
    double angularDistance = range / EarthRadius;
    double trueCourse = Math.toRadians(bearing);

double lat = Math.asin(sin(latA) * cos(angularDistance) + cos(latA) * sin(angularDistance) * cos(trueCourse));

double dlon = atan2(sin(trueCourse) * sin(angularDistance) * cos(latA), cos(angularDistance) - sin(latA) * sin(lat));

    double lon = ((lonA + dlon + Math.PI) % (Math.PI * 2)) - Math.PI;

    lat = Math.toDegrees(lat);
    lon = Math.toDegrees(lon);

    return new PointF((float) lat, (float) lon);
}

private static boolean pointIsInCircle(PointF pointForCheck, PointF center, double radius)
{
    return getDistanceBetweenTwoPoints(pointForCheck, center) <= radius;
}

private static double getDistanceBetweenTwoPoints(PointF p1, PointF p2)
{
    double R = 6371000; // m
    double dLat = Math.toRadians(p2.x - p1.x);
    double dLon = Math.toRadians(p2.y - p1.y);
    double lat1 = Math.toRadians(p1.x);

    double lat2 = Math.toRadians(p2.x);

    double a = sin(dLat / 2) * sin(dLat / 2) + sin(dLon / 2)
        * sin(dLon / 2) * cos(lat1) * cos(lat2);
    double c = 2 * atan2(Math.sqrt(a), Math.sqrt(1 - a));
    double d = R * c;

    return d;
}

public final static double AVERAGE_RADIUS_OF_EARTH_KM = 6371000;
public int calculateDistanceInKilometer(double userLat, double userLng,
                                       double venueLat, double venueLng) {

    double latDistance = Math.toRadians(userLat - venueLat);
    double lngDistance = Math.toRadians(userLng - venueLng);

    double a = sin(latDistance / 2) * sin(latDistance / 2)
        + cos(Math.toRadians(userLat)) * cos(Math.toRadians(venueLat))
        * sin(lngDistance / 2) * sin(lngDistance / 2);

    double c = 2 * atan2(Math.sqrt(a), Math.sqrt(1 - a));

    return (int) (Math.round(AVERAGE_RADIUS_OF_EARTH_KM * c));
}

@Override
protected ArrayList<BusStop> doInBackground(Void... voids)
{
    float radius2 = 1000f;
    PointF center = new PointF(currentLat, currentLong);
    final double mult = 1; // mult = 1.1; is more reliable
    PointF p1 = calculateDerivedPosition(center, mult * radius2, 0);
    PointF p2 = calculateDerivedPosition(center, mult * radius2, 90);
    PointF p3 = calculateDerivedPosition(center, mult * radius2, 180);
    PointF p4 = calculateDerivedPosition(center, mult * radius2, 270);
}

```

Fonte: Autoria própria

Figura 29. Código parcial para procura de paragens próximas via Google Maps

```
/* carregar Google maps */
private void setupMap() {
    mapFragment = (MapFragment) getFragmentManager().findFragmentById(R.id.map_nearby_stops);
    mapFragment.getMapAsync(this);
    //parametros passados para onMapReady
}

@Override
public void onMapReady(GoogleMap maps) {
    mapi = maps;
    maps.setMapType(GoogleMap.MAP_TYPE_HYBRID);
    maps.setIndoorEnabled(true);
    maps.setBuildingsEnabled(true);
    maps.setOnMarkerClickListener(this);
    maps.setOnCameraIdleListener(this);
    maps.getUiSettings().setZoomControlsEnabled(true);
}

/* */
//on Connect TO OPEN ACT Maps
@Override
public void onConnected(Bundle bundle) {
    // tem que ficar um get location
    getCurrentLocation(); //used to see and add near Stops in Maps
    getDeviceLocation(); //porque esta falahndo buscar
    mapi.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(currentLat, currentLong), cameraZoom));
    MapUtils.addNearbyStopMarkers(mapi, dbHelper, mapi.getCameraPosition().target);
}

// adicionar as infos no google maps via visualizacao de icons(markers)
public static void addNearbyStopMarkers(GoogleMap map, DBHelper dbHelper, LatLng location) {
    // adds markers to the map representing the 8 nearest bus stops to
    // the current camera position

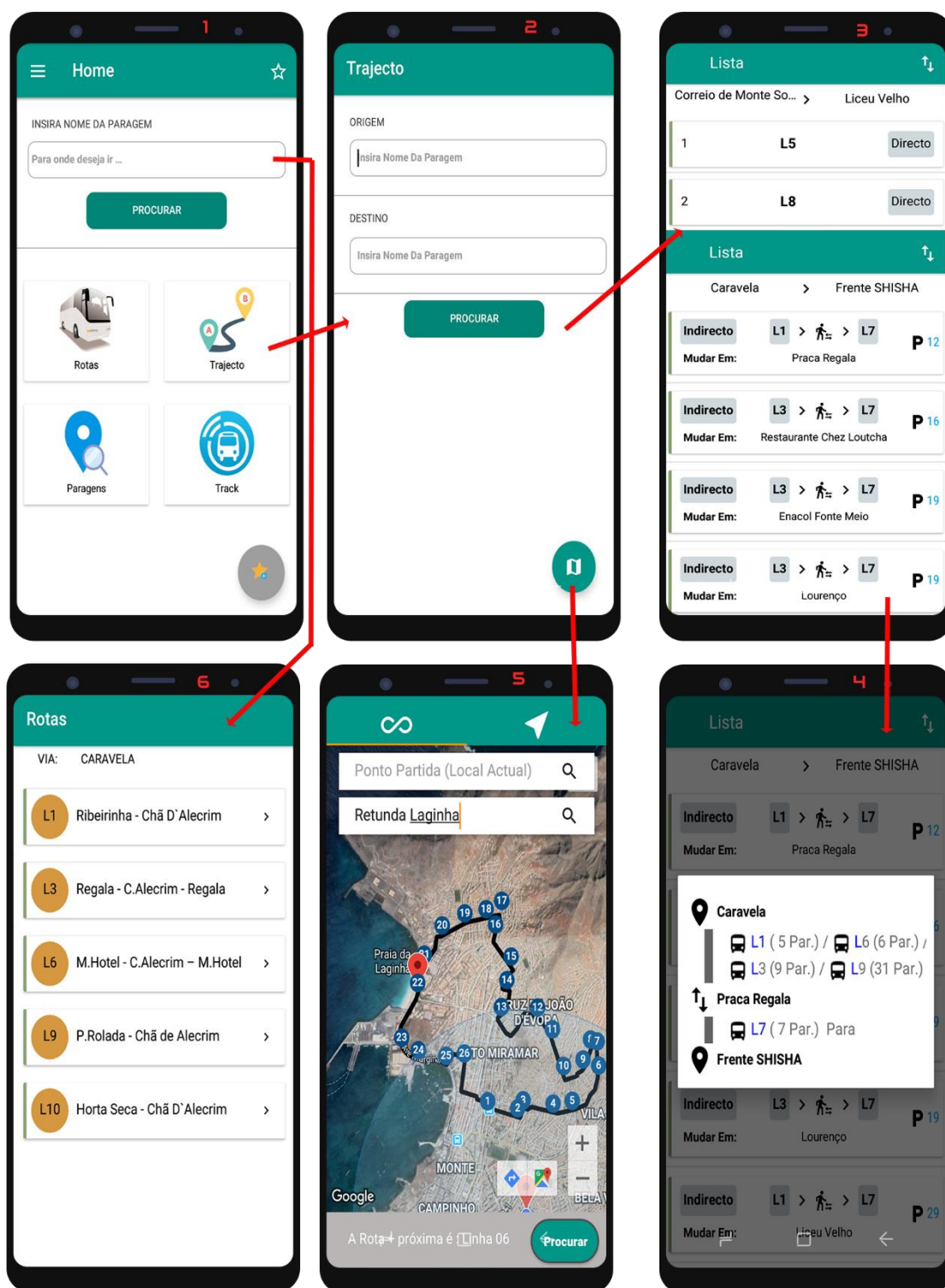
    Cursor stops = dbHelper.queryNearbyStops(location.latitude, location.longitude);
    int noStops = stops.getCount();

    stops.moveToFirst();
    for (int i = 0; i < noStops; i++) {
        addNewMarker(map, // GoogleMaps object
            stops.getString(1), // name of stop
            stops.getString(0), // stop number
            Double.parseDouble(stops.getString(2)), // latitude
            Double.parseDouble(stops.getString(3)) // longitude
        );
        stops.moveToNext();
    }
}
```

Fonte: Autoria própria

4.5.1.4 Sugestão de Rotas

Figura 30. Informações de Sugestão de Rotas



Fonte: Autoria própria

Sugestão de Rotas baseado no ponto de Destino

Aquando da sugestão de rota baseado na procura do destino, o código a seguir representa, o processo em permite o utilizador pesquisar pontos de destino e o sistema mostrar as existentes.

Funcionando a partir do momento em que o utilizador aciona o cursor no campo ‘INSIRA NOME DA PARAGEM’ na qual o utilizador começa a digitar uma palavra, e automaticamente o sistema vai reduzindo as hipóteses de palavras de destino e sugerindo as mesmas, quando o utilizador seleccionar uma palavra que corresponde a um ponto de paragem, o sistema consulta na base de dados todas as rotas que possuem esse ponto de paragens e retornando informações dessas rotas aos utilizadores, representado na figura 30, acima exposto, representado nas telas 1 e 6.

Figura 31. Código parcial para receber input e sugerir rota baseada no destino

```
// buscar informacoes de pontos de paragens e mostrar ao utilizador para
// procurar sugestoes de rotas com base destino inserido destino

db1=openOrCreateDatabase(DBNAME1, Context.MODE_PRIVATE,null);
db=stopAssistant.getReadableDatabase();
initialize();
stopName.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View arg1, int pos, long id) {

        RelativeLayout rl = (RelativeLayout) arg1;
        TextView tv = (TextView) rl.getChildAt(0);
        stopName.setText(tv.getText().toString());

    }
});

//
busDataSource = new BusDataSource(AMainDashboardActivity.this);
try {
    busDataSource.open();
} catch (SQLException e) {
    e.printStackTrace();
}
Stop myObject = null;
Stop[] ObjectItemData = new Stop[1];

myObject = new Stop("None");
ObjectItemData[0] = myObject;
myAdapter = new AutoTextCustomArrayAdapter(this, R.layout.tv_list_item_ab, ObjectItemData);
stopName.setAdapter(myAdapter);

//
stopName.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (view.getId() == stopName.getId())
        {
            stopName.setCursorVisible(true);
        }
    }
});

//
stopName.addTextChangedListener(new PassingbyAutoTextChangeListener(this));
clear.setOnClickListener(this);
search.setOnClickListener(this);
```

Fonte: Autoria própria

Sugestão de Rotas baseado na inserção de ponto de Origem e Destino

Aquando da sugestão de rota baseado na procura de ponto de origem e destino, o código a seguir representa de forma parcial, o processo em permite o utilizador inserir pontos de origem e destino e o sistema mostrar as rotas existentes, estas diretas ou indiretas.

O processo de procura baseia-se inicialmente em procurar rotas que possuem quer a origem e quer o destino, caso não encontrar, se procurará rotas indiretas, isto é que englobam pelo menos duas rotas, encontrada via interseção das mesmas, representado em termos de telas as telas 2,3,4 da na figura 30 acima exposto. Podendo ver uma melhor explicação encontrada na figura 14. Diagrama Atividade Calcular Trajeto.

Figura 32. Código parcial para sugestão de rotas via inserção de origem e destino

```
/* implementacao para captura das info de origem e destino
   e sugerir rotas directas ou indirectas
*/
try {
    busDataSource = new BusDataSource(Between.this);
    try {
        busDataSource.open();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    textViewSource.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View arg1, int pos, long id) {
            RelativeLayout rl = (RelativeLayout) arg1;
            TextView tv = (TextView) rl.getChildAt(0);
            textViewSource.setText(tv.getText().toString());
        }
    });

    textViewDestination.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View arg1, int pos, long id) {
            RelativeLayout rl = (RelativeLayout) arg1;
            TextView tv = (TextView) rl.getChildAt(0);
            textViewDestination.setText(tv.getText().toString());
        }
    });

    textViewSource.addTextChangedListener(new BetweenAutoTextChangeListener(this));
    textViewDestination.addTextChangedListener(new BetweenAutoTextChangeListener(this));

    Stop myObject = null;
    Stop[] ObjectItemData = new Stop[1];

    myObject = new Stop("None");
    ObjectItemData[0] = myObject;

    myAdapter = new AutoTextCustomArrayAdapter(this, R.layout.tv_list_item_ab, ObjectItemData);
    textViewSource.setAdapter(myAdapter);
    textViewDestination.setAdapter(myAdapter);

    searchButton.setOnClickListener(this);
    clearSource.setOnClickListener(this);
    // lytPassing.setOnClickListener(this);

    clearDestination.setOnClickListener(this);
    interchange.setOnClickListener(this);
} catch (NullPointerException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
```

Fonte: Autoria própria

Sugestão de Rotas via Google Maps e tendo em conta um ponto de origem e destino

A sugestão de rotas poderá ainda ser feita via uso do Google Maps, na qual o utilizador poderá inserir um ponto de origem (que por predefinição já reconhece a posição atual do utilizador), e um ponto de destino. Nesta implementação se sugere apenas a rota mais otimizada, sendo esta rota a mais próxima do ponto de origem e do ponto de destino, representado em termos de telas, a tela 5 na figura 30 acima exposto.

Figura 33. Código parcial sugestão de rota via GoogleMaps

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    if (ActivityCompat.checkSelfPermission(getContext(),
        Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(getContext(),
        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        return;
    }
    mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
    mMap.setMyLocationEnabled(true);
    mMap.setOnMyLocationButtonClickListener(this);
    mMap.setOnMyLocationClickListener(this);
    mMap.animateCamera(CameraUpdateFactory.zoomTo(14.0f));
    LatLng marker = BOUNDS_MOUNTAIN_VIEW2.getCenter();

    mMap.getUiSettings().setZoomControlsEnabled(true);
    mMap.setOnMyLocationButtonClickListener(new GoogleMap.OnMyLocationButtonClickListener() {
        @Override
        public boolean onMyLocationButtonClick() {
            return false;
        }
    });
}

/* Obter a rota mais próxima da origem e destino*/
public Route getNearestRoute(){
    float minDistance = 200f; //Mínimo de proximidade
    float maxDistance = 3000f; //Máximo de proximidade

    Route nearestRoute = null;
    List<Route> nearRoutesToDest = new ArrayList<>();

    float[] results = new float[1];
    //As rotas mais próximas ao destino são obtidas
    for(Route route: routes){
        for(LatLng station: route.getStations()){
            Location.distanceBetween(station.latitude, station.longitude, dest.latitude, dest.longitude, results);
            if(results[0] <= minDistance){
                nearRoutesToDest.add(route);
                break;
            }
        }
    }

    //Nas rotas mais próximas ao destino, é obtido o mais próximo da origem
    for(Route route: nearRoutesToDest){
        for(LatLng station: route.getStations()){
            Location.distanceBetween(station.latitude, station.longitude, origin.latitude, origin.longitude, results);

            if(results[0] <= maxDistance){
                maxDistance = results[0];
                nearestRoute = route;
            }
        }
    }

    return nearestRoute;
}
```

Fonte: Autoria própria

Dentro da implementação do aplicativo como se observa na Dashboard principal, há uma funcionalidade designada de **Track – Tempo Real**, que permitiria, fazer o acompanhamento das rotas e dos respetivos autocarros em tempo real e saber a distancia que se encontrava e tempo de chegada a localização atual do utilizador.

Esta funcionalidade, depende de acesso a informações de um servidor de terceiros, que aquando da ultima revisão deste trabalho se encontrava indisponível, por conseguinte, esta funcionalidade manterá neste trabalho, por via da Apêndice B, por onde se detalha alguns pormenores, e não tendo a garantia do servidor estar mais disponível, dai não inclui-lo na estrutura principal deste trabalho.

Em termos de implementação serão os acimas referidos as principais funcionalidades do aplicativo, não sendo contudo as únicas, presentes na mesma, mas para o qual não se pretende incluir referências ou explicações neste mesmo trabalho.

Dimensões do Aplicativo

Em termos de dimensões do aplicativo para instalação e uso, a mesma tem um tamanho de 9MB, e depois de instalada ficando com espaço de armazenamento ocupado de 9MB da Aplicação + 10MB de Dados, resultantes de entre outros por exemplo de descompactação dos dados da base de dados e informações do acesso aos dados do modo Online.

Segurança

Em termos da segurança, no que se refere a privacidade e acesso a permissões, o aplicativo utiliza as seguintes permissões, todos a qual é previamente pedido o seu acesso ao utilizador:

- **Armazenamento** – para permitir guardar no dispositivo os dados principalmente da base de dados SQLite.
- **Localização** – permitir aceder as funcionalidades que exigem saber a localização atual do utilizador, tais como paragens mais próximos do utilizador quer via GPS quer por meio do Google Maps.

CAPITULO VI

5 CONCLUSÃO

O serviço de transporte público constitui uma base fundamental no desenvolvimento de uma cidade, e a presença de um sistema de informação destinado aos utilizadores do mesmo, pode melhorar tanto a qualidade quanto a imagem do serviço, aumentando, assim, sua procura. E ao informar e instruir, um sistema de informação possibilita que o utilizador compreenda e utilize melhor o serviço, tornando-o realmente acessível a toda população. Sendo assim é necessário que diretamente ou indiretamente passe a considerar sistemas de informação para utilizadores como uma estratégia do transporte público coletivo.

A realização deste trabalho foi motivada pela percepção de haver uma insuficiência de informações e canais de informação sobre o sistema de transporte público na cidade, a qual os utilizadores podem se defrontar, e sabendo-se que este serviço é amplamente utilizado na cidade do Mindelo, surgiu desta problemática a perspectiva de estudar formas de facilitar o acesso a estas informações de maneira rápida e eficiente, disponibilizando para isso um aplicativo para dispositivos móveis, oferecendo informações desde a localização das rotas e pontos de paragens, até a determinação de rotas ótimas, entre a origem e destino informados pelos utilizadores.

Tendo em conta os objetivos propostos inicialmente para este trabalho se considera ter atingido os mesmos, com bons resultados alcançados, utilizando ferramentas tecnológicas diversos ao dispor de todos, e tornando viável e sem investimentos maiores implementar uma solução sob forma de um aplicativo dispositivos moveis, que constitui hoje uma dos principais canais onde pessoas procuram informação e portando podendo ser utilizado por pessoas que estejam interessadas.

Espera-se que a implementação da mesma possa contribuir para a melhoria de prestação desse serviço e contribuindo para a inovação tecnológica no cidade e no país.

5.1 Resultados Obtidos

Tendo em conta a ideia com que se propôs a analisar e a propor uma solução, se considera que os resultados desejados foram atingidos, havendo margem para melhorar em trabalhos futuros, mesmo que a solução no momento seja desenvolvido de trabalho de conclusão de estudos, sendo uma proposta por enquanto de protótipo e de experimento para demonstração das tecnologias estudadas.

As contribuições deste projeto se refere ao que se apresentou uma solução para a problemática da falta de informação sobre as rotas, pontos de paragens nos trajetos dos autocarros, ao criar mapas das linhas que permite ter uma visualização mais detalhada das rotas, e se poder se situar melhor em relação ao uso deste serviço da melhor forma possível.

Uma vantagem alcançada com este aplicativo é que funciona quer em modo online quer em modo offline – com restrições neste último, mas que não impede o seu uso.

Considera-se, portanto, que o trabalho desenvolvido atendeu significativamente às expectativas iniciais, inclusive superando-as em alguns casos.

5.2 Dificuldades Encontradas

As dificuldades tiveram a ver com a análise e a busca dos dados para se realizar e alimentar o sistema, visto que estas foram recolhidas em um website do operador Transcor, que aquando da última revisão deste trabalho não mais se encontrava disponível, portanto tornando difícil atualizar os dados para o sistema.

De referir, das implementações efetuadas, há uma funcionalidade que permitiria o acompanhamento dos autocarros em Tempo real, mas que devido a indisponibilidade dos dados onde estes eram obtidos, não mais é possível garantir o funcionamento da mesma.

5.3 Desenvolvimentos futuros

Sendo a aplicação desenvolvida apenas um protótipo é visto que há uma abrangência de funcionalidades a serem exploradas e melhoradas, e com novas tecnologias.

Neste sentido uma destas funcionalidades a serem implementadas seria, a de se poder aproveitar e transferir os dados diretamente para a ferramenta já conhecida, completa e consolidada como o Google Maps, permitindo que as informações estejam disponíveis a nível global e independente da plataforma e para todos, contanto para isso o interesse das agências que disponibilizam o serviço de transporte público na cidade, para que façam a parceria necessária com o Google Transit.

De se considerar criar uma página web para fornecer informações sobre as rotas de transporte urbano, e não ficar apenas no nível do smarphone Android.

Quiçá também criar um aplicativo também na plataforma iOS.

Como continuação deste trabalho, espera-se também que o contributo deixado por este projeto possa ser utilizado para futuras melhorias da solução apresentada, visto ser ideia do autor que o código fonte desta solução possa ser Open Soure sob a licença MIT, para quem assim por ventura achar útil.

REFERÊNCIAS BIBLIOGRÁFICAS

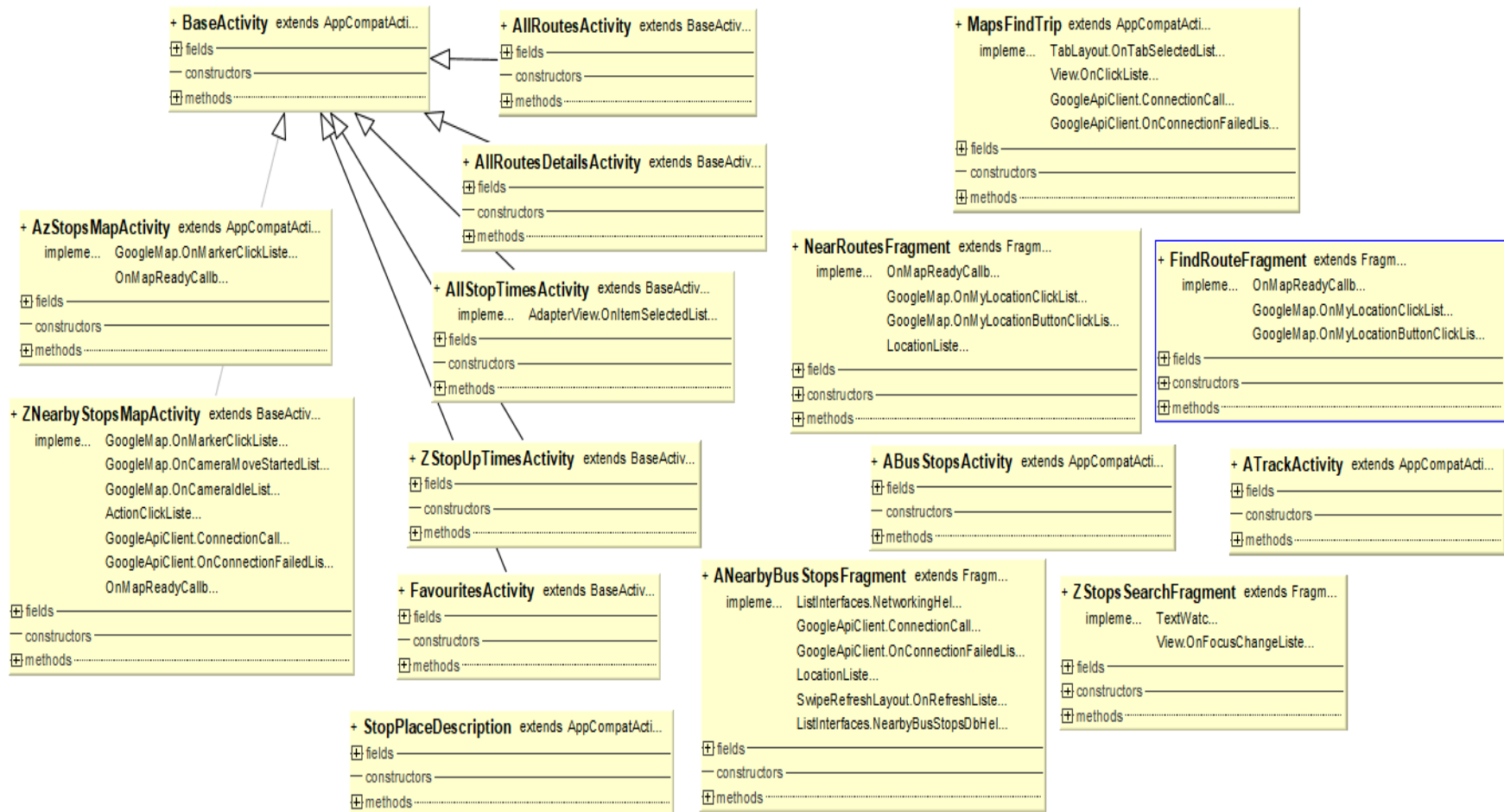
- Ambros, L. (2013). *Diferença entre aplicativos nativos, híbridos e mobile web*. Obtido de luisaambros: <https://www.luisaambros.com/blog/diferenca-entre-aplicativos-nativos-hibridos-e-mobile-web-apps/>
- ARME. (07 de 07 de 2020). *Relatório de indicadores estatísticos das comunicações electrónicas – 1º trimestre de 2020*. Obtido em 15 de 07 de 2020, de ARME: <https://www.arme.cv/index.php/component/jdownloads/send/49-relatorios-arme/765-relatorio-de-indicadores-estatisticos-das-comunicacoes-electronicas-1-trimestre-de-2020>
- Developers, A. (2020). *Conheça o Android Studio*. Obtido em 05 de 03 de 2020, de Developers Android: <https://developer.android.com/studio/intro/index.html?hl=pt-br>
- Fernandes, R. G. (2007). *Componentes gráficos para um sistema de informação visual em terminais de integração metrô-ônibus*. Obtido em 05 de 03 de 2020, de unb: https://repositorio.unb.br/bitstream/10482/2005/1/Dissert_Reverson%20Geraldo%20dos%20Anjos%20Fernandes.pdf
- Glauber, N. (2015). *Dominando android: do básico ao avançado*. Obtido em 05 de 03 de 2020, de books.google: <https://books.google.cv/books?id=E6LmCgAAQBAJ&printsec=frontcover#v=onepage&q&f=false>
- Google. (2020). *Google Maps*. Obtido em 05 de 03 de 2020, de Google: <https://www.google.com/intl/pt-BR/maps/about/>
- Hammerschmidt, R. (13 de 07 de 2015). *Linha do Tempo: por dentro da evolução do Android*. Obtido em 13 de 02 de 2020, de Tecmundo: <https://www.tecmundo.com.br/android/82344-linha-tempo-dentro-evolucao-do-sistema-android.htm>
- INE. (28 de 02 de 2020). *Estatísticas dos Transportes – 4º Trimestre de 2019*. Obtido em 28 de 03 de 2020, de INE: <http://ine.cv/wp-content/uploads/2020/02/publicacao-4-trimestre-2019-003.pdf>
- Lanzoni, C., Scariot, C., & Spinillo, C. G. (2011). *Sistema de informação de transporte público coletivo no Brasil: algumas considerações sobre demanda de informação dos*

- usuários em pontos de parada de ônibus*. Obtido em 03 de 03 de 2020, de Infodesign: <https://www.infodesign.org.br/infodesign/article/view/114>
- Marques, H. N. (1998). *Um sistema de informações para usuários de transporte coletivo em cidades de medio porte*. Obtido em 25 de 03 de 2020, de teses.usp.br: https://www.teses.usp.br/teses/disponiveis/18/18137/tde-02052007-212934/publico/Ms_HMarq.pdf
- Neka, E. (6 de 12 de 2015). *Geolocalização no desenvolvimento de aplicativos mobile é*. Obtido em 05 de 03 de 2020, de yeePLY: <https://pt.yeeply.com/blog/geolocalizacao-no-desenvolvimento-de-aplicativos-mobile-e-importante/>
- Ogliari, R. d., & Brito, R. C. (2014). *Android - do básico ao avançado*. Obtido em 13 de 02 de 2020, de livrariacultura: <http://site.livrariacultura.com.br/imagem/capitulo/15056106.pdf>
- Pressman, R. (2011). *Engenharia de Software 7ª Edição - Uma Abordagem Profissional*. Obtido em 05 de 03 de 2020, de scribd: <https://pt.scribd.com/doc/257550621/Roger-S-Pressman-Engenharia-de-Software-7-Edicao-Uma-Abordagem-Profissional>
- Silva, D. M. (2020). *Sistemas Inteligentes No Transporte Publico Coletivo Por Ônibus*. Obtido em 05 de 03 de 2020, de ufrgs: <https://www.lume.ufrgs.br/bitstream/handle/10183/3134/000287914.pdf>
- SQLITE. (2020). *About. SQLite*. Obtido em 05 de 03 de 2020, de SQLite: <https://www.sqlite.org/about.html>

APÊNDICE

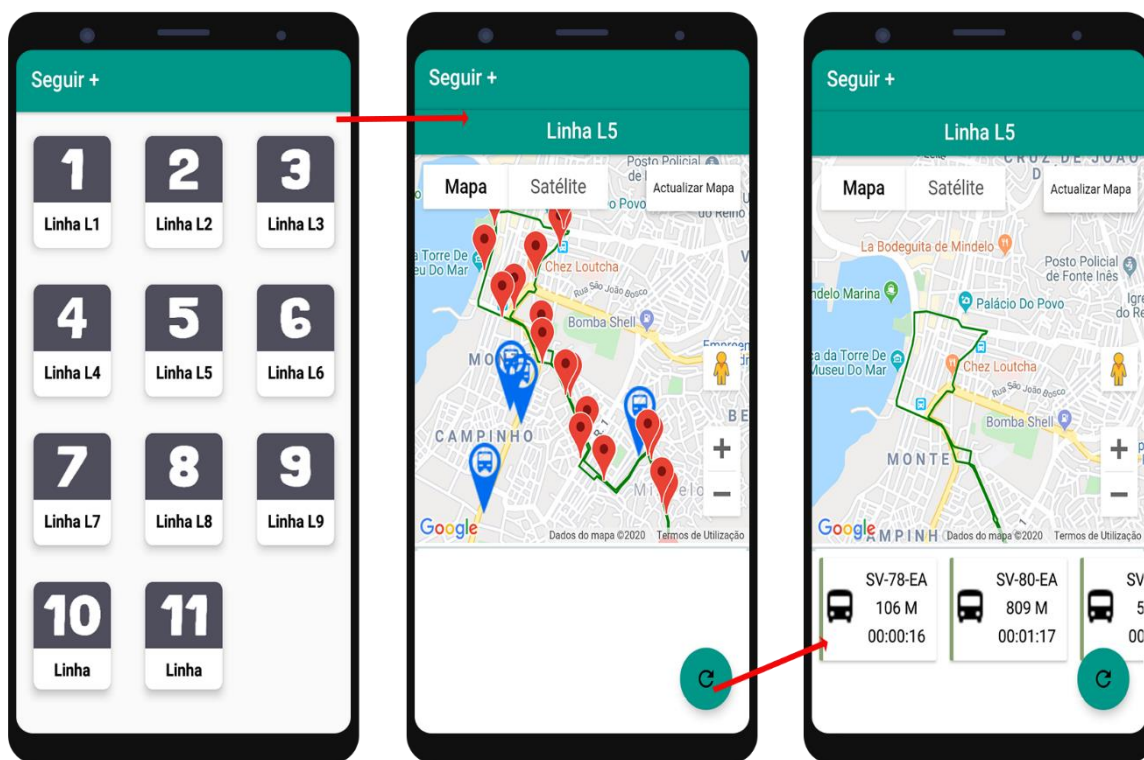
APÊNDICE A – Diagrama de Classes principais da aplicação

Figura 34. Diagrama de Classes principais da aplicação



APÊNDICE B – Track – Tempo Real

Figura 35. Visualização de Rotas em Tempo Real



Fonte: Autoria própria

Em termos de código para implementação da funcionalidade temos como necessários os principais dados: **Gps_DataHora**; **Gps_CodAutocarro**; **Matricula**; **Gps_CodLinha**; **Gps_CodTurno**; **Gps_Latitude**; **Gps_Longitude**; **Gps_Velocidade**;

A codificação parcial da mesma, se encontra na figura a seguir, e representa o uso de uma Webview (que permite visualizar uma página de um website dentro de num página/layout de uma aplicação Android, o uso de um Gson (Gson é uma biblioteca Java de código aberto para serializar e desserializar objetos Java para JSON). E dos cálculos necessários.

Figura 36. Código parcial implementação do Track Tempo Real

```
/* Carregar uma Webview para carregar infos de uma website num layout android */

webView = findViewById(R.id.webView);
rootView = findViewById(R.id.root_view);
recyclerView = findViewById(R.id.my_recycler_view);
webView.loadUrl(url);
webView.setInitialScale(1);

/* */

/* necessarios para obter dados da api via javascript para calcular estimativas de tempo */
public void getConsoleGson() {
    try {
        String teste = all;
        String beta2 = "}]";
        String autobbb = (teste.substring(teste.indexOf("autocarros"), teste.lastIndexOf(beta2) + 2).trim());
        String last2 = autobbb.substring(autobbb.length());
        WebViewAutoModel[] reviews = new Gson().fromJson(autobbb, WebViewAutoModel[].class);
        asList = Arrays.asList(reviews);

        recyclerView.setHasFixedSize(true);
        LinearLayoutManager layoutManager2 = new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false);
        GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 2, GridLayoutManager.HORIZONTAL, false);
        recyclerView.setLayoutManager(layoutManager2);

        // specify an adapter
        mAdapter = new WebViewAdapter(getApplicationContext(), asList);
        recyclerView.setAdapter(mAdapter);
        //Toast.makeText(getApplicationContext(), String.valueOf(teste), Toast.LENGTH_SHORT).show();

    } catch (Exception e) {
        e.getMessage();
    }
}

/* necessarios para obter dados da api via javascript para calcular estimativas de tempo */

/* necessarios para calcular a distancia e tempo para percorrer distancia ate localizacao do utilizador */
public void getDistance() {

    LatLng endLatLng;
    LatLng bus;
    Double latitude;
    Double longitude;
    int speedInKm;
    Long seg;

    latitude = Double.valueOf(childListget.get(position).getGps_Latitude());
    longitude = Double.valueOf(childListget.get(position).getGps_Longitude());
    bus = new LatLng(latitude, longitude);
    //endLatLng = new LatLng(16.878664, -24.982920);
    endLatLng = new LatLng(myLastLocation.getLatitude(), myLastLocation.getLongitude());
    speedInKm = Integer.parseInt(childListget.get(position).getGps_Velocidade());
    double distance = SphericalUtil.computeDistanceBetween(bus, endLatLng);
    String metros = String.valueOf(Math.round(distance));
    String segundos = String.valueOf(Math.round(distance / ((0.277778 * speedInKm))));
    seg = Long.parseLong(segundos);
    int hours = (int) ((seg) / 3600);
    int minutes = (int) (((seg) % 3600) / 60);
    int seconds = (int) ((seg) % 60);
    @SuppressWarnings("DefaultLocale") String timeString = String.format("%02d:%02d:%02d", hours, minutes, seconds);
    int km2 = (Integer.parseInt(metros) / 1000);
    int m2 = ((Integer.parseInt(metros) % 1000) / 100);
    String metrosx = (metros + " M");
    @SuppressWarnings("DefaultLocale") String distancekm = ((String.format("%01d.%01d", km2, m2) + " KM"));
}
```

Fonte: Autoria própria